Neurocomputing 506 (2022) 96–116

Contents lists available at ScienceDirect

Neurocomputing

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

# A review on anchor assignment and sampling heuristics in deep learning-based object detection

Xuan-Thuy Vo, Kang-Hyun Jo *

*Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 44610, South Korea*

ARTICLE INFO

*Article history:*
Received 9 April 2022
Revised 22 May 2022
Accepted 12 July 2022
Available online 16 July 2022

*Keywords:*
Object detection
Deep learning
Convolutional neural networks (CNNs)
Anchor assignment
Sampling heuristics
Transformer-based object detection

ABSTRACT

Deep learning-based object detection is a fundamental but challenging problem in computer vision field, has attracted a lot of study in recent years. State-of-the-art object detection methods rely on the selection of positive samples and negative samples, i.e., called sample assignment, and the definition of a useful set for training, i.e., called sample sampling heuristics. This paper presents a comprehensive review of the advanced anchor assignment and sampling approaches in deep learning-based object detection. Each problem is classified and analyzed systematically. According to the problem-based taxonomy, we identify the advantages and disadvantages of each problem in-depth and present open issues regarding the current methods. Furthermore, this paper also reviews the new trends in solving object detection that has not been discussed during the last two years. To track the latest research, a webpage related to the above problems is provided, which is available at https://github.com/VoXuanThuy/ObjectDetectionReview.


## 1. Introduction

Object detection served as a primary process for high-level tasks such as multiple object tracking [1–6], skeleton-based action detection [7,8], action detection [9,10], person search [11–13], facial landmarks detection [14], and human pose estimation [15–17], has been widely used in many real-world applications, e.g., video surveillance systems [18–21], autonomous vehicles [22–25], and vision robotics [26–30]. The aim of object detection is to identify what objects are presented in an image from pre-defined categories such as person, car, zebra, etc., and determine where objects are located through rectangular bounding boxes (spatial location).

The first generation of the object detector were solved by traditional machine learning techniques with representative methods such as dimensionality reduction [31–33], principal component analysis [34], improved ensemble systems [35], linear discriminant analysis [36–39], optimization [40–43], constrained learning algorithms [44–46], hybrid learning methods [47], and shallow neural networks [48–52]. These techniques heavily rely on hand-crafted features and linear classifiers. The popular approaches in this generation are the Histogram of Oriented Gradients (HOG) [53], and Deformable Part Model (DPM) [54]. HOG is a feature descriptor that computes local intensity histograms of gradient orientation in a dense grid of image cells to enhance the scale-invariant feature transform descriptors [55,56], and shape matching [57]. DPM extends HOG orientation histograms, improving detection performance by introducing discriminative training (mining hard negative samples during training) and multi-scale deformable parts. As the limitation of hand-crafted features, the performance of object detection became saturated during 2010–2012. After AlexNet [58] proposed the dominant work in 2012, the revolution of deep convolutional neural networks has started to solve complex problems in computer vision, and object detection also has been dominated. The current generation leverages the object detection model to be effective and efficient via Convolutional Neural Networks (CNNs). This innovative technology has brought remarkable improvement in terms of accuracy and computational cost. Specifically, the state-of-the-art detectors based CNNs achieved approximately 0.89 mAP (mean Average Precision) on the benchmark dataset PASCAL VOC while DPM-based hand-crafted features only achieved 0.34 mAP.

In recent years, deep learning-based object detection (deep object detection) has been dominated by anchor-based detectors or anchor-free detectors, predicting the classification scores and regression offsets for the set of anchors (candidate boxes). To train the detection model, we should define the classification and regression targets for each anchor. This is called *anchor assignment* in deep object detection. In this paradigm, anchors are assigned as positive or negative samples according to a certain criterion. Since object detection performance is sensitive to the definition of posi-

* Corresponding author.
*E-mail addresses:* xthuy@islab.ulsan.ac.kr (X.-T. Vo), acejo@ulsan.ac.kr (K.-H. Jo).

https://doi.org/10.1016/j.neucom.2022.07.003
0925-2312/

tive and negative samples, many studies invest much effort into advanced algorithmic approaches. In nature, most anchor boxes are labeled as negative samples (background class) because several spatial locations in each image contain objects. It leads to an imbalance problem between negative and positive samples in object detection. When not figured out, the model heavily pays attention to negative samples. This problem directly degrades the final detection performance. The critical solution is to select a subset of negative and positive samples to train detectors efficiently. This procedure is called *sampling heuristics* in object detection. In recent years, the object detection community has addressed this problem in many aspects.

In this paper, we describe deep learning-based object detection in terms of anchor assignment, sampling heuristics, and recent trends of the object detection in systematic manners. Three components are identified and classified in the problem-based taxonomy to study the problem and the solution. The systematic taxonomy associated with the list of the existing papers for each problem is shown in Fig. 1, based on its goal, solutions, and structural networks.

### 1.1. Comparison with previous reviews

In recent years, many detailed deep object detection reviews based on a taxonomy have been presented in [59–62]. These reviews described the milestone object detection, benchmark datasets and metrics, detection frameworks, feature extractors, main blocks in detectors, and state-of-the-art methods from the 1990s to 2019. Zhao et al. [63] proposed a review for object detection approaches that treat challenging problems such as occlusion, clutter, and feature scales in RCNN [64] and its variants. Oksuz et al. [65] analyzed imbalance problems in object detection and proposed open research issues for each problem. These problems are categorized into four groups: foreground-background or foreground-foreground imbalance, object scale imbalance, spatial bounding box imbalance, and objective imbalance due to multi-task learning. Unlike existing surveys, we classify state-of-the-art detectors according to anchor assignment and sample sampling components and provide detailed analysis for each classified method. To the best of our knowledge, there is no prior method discussing these two problems in object detection literature.

Some surveys provided the summarized literature for specific object detection, such as vehicle detection [66,67], face detection [68,69], and pedestrian detection [70–72]. Dollar et al. [70], Cao et al. [71], and Hosang et al. [72] present an in-depth analysis of feature extractors from hand-crafted features to deep learning. Cao et al. [71], and Hosang et al. [72] further discuss some problems in pedestrian detection such as occlusion, scale variance, and domain adaptation. These surveys concentrate on the specific class and do not analyze the core components related to the input property of generic object objection.

### 1.2. Scope

The main purpose of this paper is to provide a comprehensive review of anchor assignment, sampling heuristics in CNN-based object detection and Transformer-based object detection, and present a problem-based taxonomy in a high-level view and state-of-the-art detectors in the last two years. Our review analyzes the advantages and disadvantages of each method, the similarities and differences between a problem-based taxonomy. We hope readers can understand current research from a general perspective and identify open research issues in the future.

Reviewing the generic object detection in cornerstone is out of the scope of this paper. We only introduce basic knowledge on widespread object detection to make researchers familiar with the concept of object detection and its components. To explore the milestone object detection, we defer to the recent surveys [59–61,63] the detailed knowledge of the detection frameworks.

The main contributions are summarized as follows:

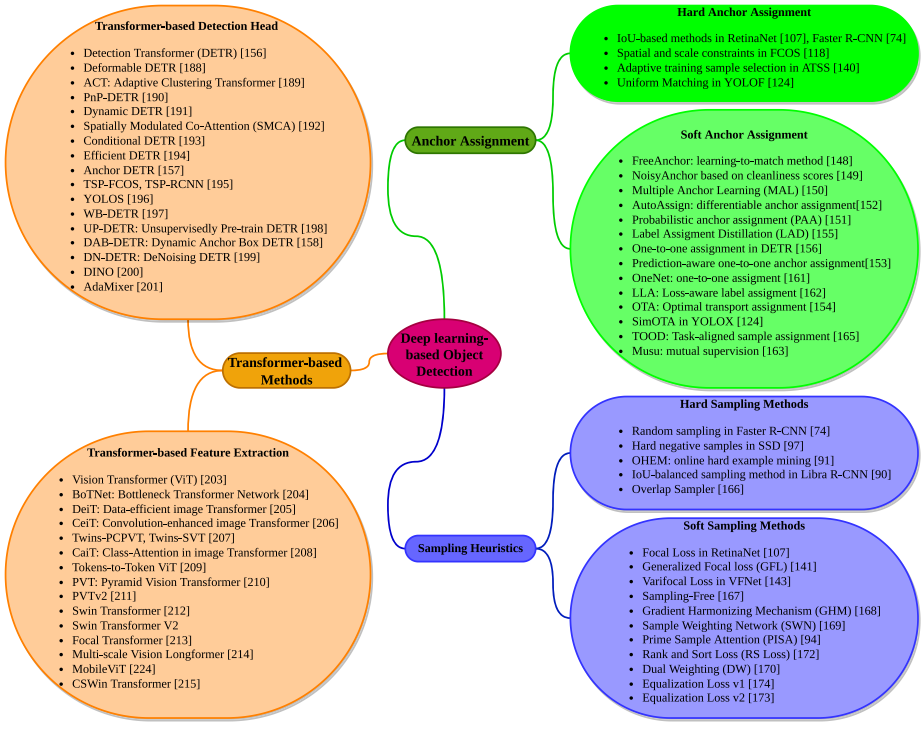



**Fig. 1.** The problem-based taxonomy of the deep-learning based object detection related to anchor assignment, sampling heuristics, and recent trends in object detection methods.

1. A detailed literature review for the existing methods about anchor assignment and sample sampling is investigated and analyzed in a taxonomy. We identify a definition of each component, challenging problems in existing methods and their solution, and a comprehensive comparison of all methods.
2. Open research direction in both anchor assignment and sampling heuristics is discussed.
3. We review the state-of-the-art object detection in the last two years that complement recent surveys [59–61,63]. We also provide paper literature related to the above problems via the repository webpage.

In the following, we introduce an overview architecture and crucial components for deep learning-based object detection in Section 2. A comprehensive review of anchor assignment and sampling methods is discussed and analyzed in Section 3 and 4, respectively. Section 5 provides new trends in solving deep learning-based object detection based on vision Transformer. Moreover, we list future potential research directions with respect to the problem-based taxonomy in Section 6. The overview flowchart of the paper is sketched in Fig. 2.

## 2. Preliminary on object detection

### 2.1. Categories of the object detection method

In this subsection, we briefly describe all components in object detection and types of object detection. Based on prior knowledge (anchor generation, regression variables), there are two types of object detection: anchor-based object detection and anchor-free object detection.

#### 2.1.1. Anchor-based object detection

Many advanced detectors have been dominated by anchor-based methods categorized into two groups: two-stage object detection and one-stage object detection. **Two-stage object detection.** The family of RCNN [64,73,74] has been pioneering works in two-stage anchor-based methods. RCNN applies a selective search algorithm [75] for generating many region proposals (about 2000 region proposals for each image). Then, the region-wise CNNs extract features and classify each region proposal using SVM. Instead of forwarding the region proposals to CNNs, Fast R-CNN directly feeds the input image to the CNN to create the feature map. Then, they generate the region proposals from the feature map using selective search and reshape them into a fixed size $(7 \times 7)$ utilizing the RoI pooling layer. The classification scores and regressed bounding box for each proposal are predicted through stacked fully connected layers from pooled features. Although Fast R-CNN reduces training and testing time, region proposals generated by the selective search are a matter in Fast R-CNN architecture since selective search is a slow and time-consuming step. To overcome this problem, Faster R-CNN introduces an *anchor generation* mechanism to create dense anchor boxes (prior bounding boxes). In general ways, multiple anchors of different scales and aspect ratios are placed to each feature map location to encompass all objects with various sizes and shapes. Faster R-CNN includes two stages: Region Proposal Network (RPN) and region-wise RCNN. In the first stage, RPN uses two CNN sub-networks to predict objectness scores and regressed offsets from the set of anchor boxes. The main goal of RPN is to reduce the number of negative samples by eliminating low-quality bounding boxes via Non-Maximum Suppression (NMS), i.e., the suppressed boxes have low objectness scores. To train RPN, anchor boxes are separated into two sets: a set of negative samples and a set of positive samples. The bounding box regression only refines bounding boxes of positive samples. In the second stage, the RCNN network further processes filtered bounding boxes in RPN to get final detection results in which RoI/RoIAlign is used to crop refined bounding boxes before feeding to classification and regression networks. Inspired by Faster R-CNN, many improved object detection methods are proposed such as network design [76–83], attention blocks [84–90], training loss and sampling heuristics [91,92,90,93,94]. In recent years, object detectors have achieved state-of-the-art performances based on two-stage methods on challenging benchmark datasets such as MS-COCO [95], Pascal VOC [96].

**One-stage object detection.** Without RPN, one-stage detectors directly predict object classification scores and regression offsets at each spatial location from assigned dense anchor boxes, which balances accuracy and speed. The representative methods of one-stage detectors are SSD [97] and its variants [98–101], YOLO family [102–106], and RetinaNet [107]. SSD places anchor boxes on multiple feature map with different scale and then, directly predicts object categories and box offsets. RetinaNet improves the one-stage network in many aspects, such as applying a feature pyramid [108] for solving scale imbalance in which anchor boxes are densely tiled on each feature map; proposing Focal loss to handle foreground/background imbalance; and designing classification and regression sub-networks. Nowadays, one-stage object detectors achieve similar performance with two-stage methods but higher testing speed than two-stage detectors.

#### 2.1.2. Anchor-free object detection

Recently, many researchers have great attention to anchor-free methods due to their high efficiency and flexibility. Anchor-free object detections directly output object categories and bounding box regression without designing anchor boxes. It is split into two groups: key-point based object detection and center-based object detection.

**Key-point based object detection.** Key-point methods locate bounding boxes by predicting important key-points, e.g., a pair of top-left and bottom-right corners [109], center points [110], key-point triplets [111], and extreme points [112] on objects. And then,

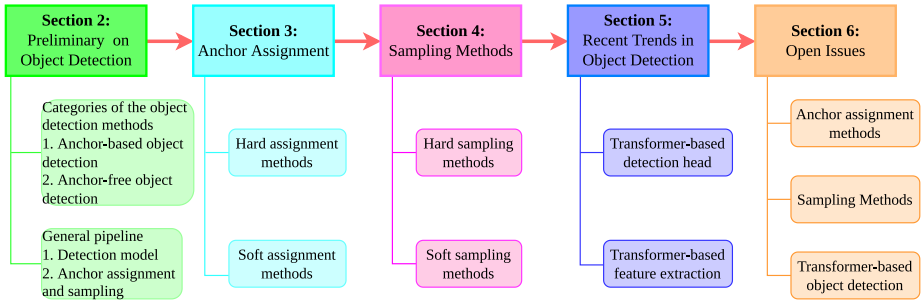



**Fig. 2.** The organization of the paper.

these methods group learned key points to generate the final box by employing associative embedding [113]. RepPoints [114] form bounding boxes as a set of sample points and then learn and arrange representative points accordingly. Although key-point methods achieve on-par results with anchor-based methods, they require longer training to converge the model.

**Center-based object detection.** Center-based methods consider the center point or center region of the object as a strict criterion to define positive and negative samples. During training, these methods regress the distance offsets from positive samples to four sides of the object boundary. YOLO [102] separates the input image into an $S \times S$ grid. If the center of an object belongs to a grid cell, that grid cell is used to detect that object. GA-RPN [115] determines the pixels inside the center region of a ground truth box as positive samples and then predicts the anchor location and shape. FSAF [116] defines the center region of an object as positive according to the prediction of the anchor-free branch with a feature selection module integrated into the detection head. FoveaBox [117] defines the region inside the middle part of an object as a positive area and then predicts four distance offsets from each cell inside the positive area to the object boundary. FCOS [118] considers anchor boxes as anchor points, eliminating hyperparameter selections of anchor boxes such as how many anchor boxes are tiled per spatial location, scale, and aspect ratio. If an anchor point falls into the object region, this point is assigned as a positive sample and utilized to regress distance offsets from this point to each side of the object boundary.

### 2.2. General pipeline

The goal of object detection is to solve multi-task learning including classification and localization tasks. The common pipeline of the object detection network is illustrated in Fig. 3, which has two components: detection model and anchor assignment/sampling. The detection model consists of:

- **Backbone Part.** Given the input image $I \in \mathbb{R}^{H \times W \times 3}$, the backbone network extracts informative features through the popular CNNs such as VGG [119], ResNet [120], MobileNet [121], where $H, W$ are height and width of the image spatial dimension.
- **Neck Part.** This network constructs multi-level feature maps with different scales to solve scale imbalance in detection. There are many methods to improve the neck parts such as FPN [108], PANet [122], Libra FPN [90], NAS-FCOS [118], NAS-FPN [123].
- **Classification and Localization Parts.** These parts are called detection head, which uses extra convolutional layers to output object categories and offsets for each assigned anchor.

The anchor assignment and sampling include:

- **Anchor Box Set:** $\mathscr{A}$ - The set of anchor boxes stands for a set of pre-defined bounding boxes (prior boxes) and $a_i \in \mathscr{A}$ is one anchor box. In most detection methods, multiple anchor boxes with different scales and aspect ratios are placed on each spatial location of the feature map to cover various objects in the image $I$. Fig. 4 illustrates anchor generator on the pyramid level $i$ with different scales and aspect ratio. In the prevalent detector RetinaNet [107], and RPN in Faster R-CNN [74], the areas of the base anchor boxes are set to $\{8^2, 16^2, 32^2, 64^2, 128^2\}$ corresponding to the feature level from $P_3$-$P_7$. On each location of the feature level $P_i \in \{P_3, P_4, P_5, P_6, P_7\}$, they place anchor boxes with three scales $\{2^0, 2^{1/3}, 2^{2/3}\}$ and three aspect ratios $\{1:1, 2:1, 1:2\}$ . Accordingly, there are 9 anchor boxes per feature location. In total, there are $H_i \times W_i \times 9$ anchor boxes for one feature map where $H_i, W_i$ are height and width of the feature level $i$, respectively.
- **Ground Truth Set:** $\mathscr{G}$ - It is a set of ground truth bounding boxes and class labels. Each element of this set is tuple $(g_j, l_j)$ where $g_j \in \mathbb{R}^4$ indicates one ground truth box with four coordinates and $l_j \in \mathbb{R}^C$ is the enumeration of the pre-defined number of classes $C$ in datasets.
- **Model's Feedback.** The learning status of classification and localization subnetworks is used as matching cost to perform the anchor assignment task.
- **Anchor Assignment.** The anchor box set is assigned to a set of positive samples and a set of negative samples.
- **Sampling Heuristics.** This procedure is to select a subset from the assigned set of anchor boxes.
- **Positive sample set -** $\mathscr{P}$**.** The assigned and sampled anchor boxes inside the positive set are close to the ground truth box location.
- **Negative sample set -** $\mathscr{N}$**.** The classification subnetwork is designed to classify this set as a background class. And, this set is not joined in performing the localization task.

## 3. Anchor assignment

Assigning the anchor boxes into the positive and negative sets is necessary processing before training detectors, which directly

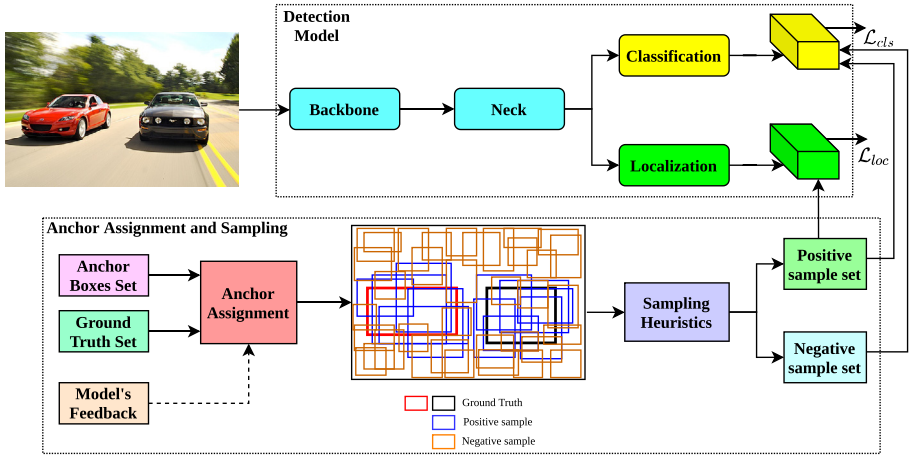



**Fig. 3.** The common pipeline of the general object detection network. The pipeline includes two main components: Detection model and Anchor assignment & sampling. $\mathscr{L}_{cls}$ indicates classification loss. $\mathscr{L}_{loc}$ denotes localization loss.

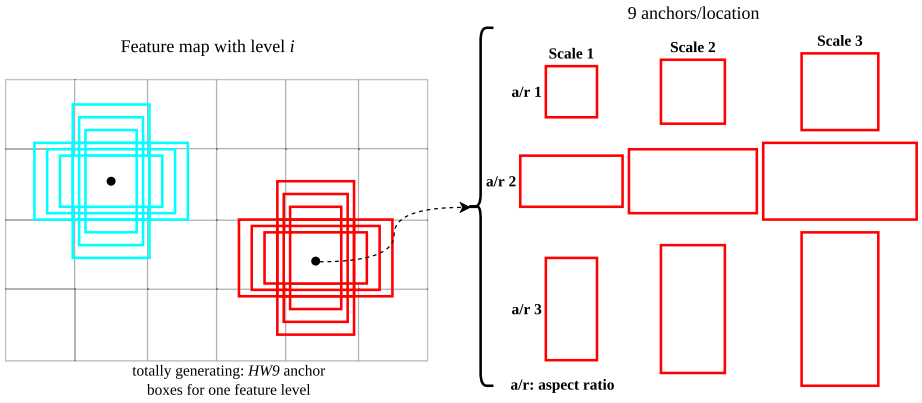



**Fig. 4.** Anchor box generator on the feature map with level $i$. The base anchor box has an area of $2^i$.

affects the detection performance. In detection literature, anchor assignment is grouped into two: (i) hard assignment methods and (ii) soft assignment methods. The difference between hard and soft assignment methods relies on the input variables of the desired algorithm to separate the anchor set into positive and negative sets according to different criteria. Hard methods compute the localization cost from the ground truth and anchor sets to perform the assignment based on hard criteria (pre-defined thresholds). Soft methods automatically separate positive and negative samples for a ground truth box according to the model's feedback during training without assuming any threshold. The matching cost of the soft assignment methods is computed by linearly combining classification and localization costs where signals are taken from learning status. This strategy produces richer semantic information for the assignment while hard methods only consider localization cost that ignores the context of objects and treat positive and negative separations independently. We summarize the key procedure and comparative performance of hard and soft assignment methods in Table 1 and Table 2, respectively. Here, FPN indicates Feature Pyramid Network [108], DE means Dilated Encoder [124], and 1× denotes 12 epochs configured during training.

### 3.1. Hard assignment methods

The hard assignment is a widely-used method in object detection. Most anchor-based detectors usually use an IoU-based method that computes the IoU (Intersection of Union) between anchor set $\mathscr{A}$ from different pyramid levels and ground truth box set $G$. For each $g_j \in G$, these detectors define a hard IoU threshold to separate positive and negative samples as follows:

$$l_{ij} = \begin{cases} 1 & \text{if } IoU(a_i, g_j) \geqslant t_p \\ 0 & \text{if } IoU(a_i, g_j) < t_n \\ -1 & \text{otherwise}, \end{cases} \tag{1}$$

where $t_p, t_n$ are IoU thresholds to define positive and negative samples for a ground truth box $g_j$, respectively. If $IoU(a_i, g_j)$ is larger than $t_p$, this anchor box is assigned as positive sample $a_i \in \mathscr{P}$ and labeled as $l_{ij} = 1$ during training. An anchor is assigned as negative sample $a_i \in \mathscr{N}$ if $IoU(a_i, g_j)$ is less than $t_n$, labeled as $l_{ij} = 0$. Otherwise, an anchor is unassigned if $IoU(a_i, g_j) \in [t_n, t_p)$ and it is ignored during training. The reason for the unassigned anchors with $IoU(a_i, g_j) \in [t_n, t_p)$ is to make the separation boundary between a set of positive samples and a set of negative samples more clear and avoid ambiguous learning originating from uncertainty problems such as occlusion, ambiguities, blur, shadow, and complex scenes. These unassigned anchors are called hard samples that generate high loss during training. If these anchors are used during training, object's coordinates and categories are not clear enough because of uncertainty. Therefore, existing detectors ambiguously identify the exact object locations and classes from the assigned anchors. As a result, detectors yield mislocalized and misclassified bounding boxes. Although the models predict high probability scores for the classification task, the box predictions do not satisfy high accuracy requirement. Therefore, it directly affects the overall performance. RetinaNet [107] sets $t_p = 0.5$ and $t_n = 0.4$ for training detection model. Fig. 5(a) shows the definition of positive samples and negative samples in RetinaNet detector. RPN in Faster R-CNN [74] adopts $t_p = 0.7$ and $t_n = 0.3$ as assigning criterion. In literature, the IoU-based method is the simple but effective strategy, applied to many detectors such as two-stage detectors [74,79,108,125–135] and one-stage detectors [97,103,104,107,136–139]. Even though IoU-based anchor assignment achieves significant improvements, they has limitations:

- For slender objects, objects with irregular shapes, occluded objects, and ambiguous objects, the anchors assigned as positive samples contain the noisy background (noisy anchors), ambiguous information for learning. Moreover, anchors with small IoU scores contain informative features for classifying and localizing objects. These factors bring harmful gradients to detection models, i.e., ambiguous learning (hard to learn and generate large losses during training).
- Scale imbalance on positive anchors is identified. This imbalance originates from the number of positive samples assigned for large ground truth boxes. In the natural, large ground truth boxes produce more positive samples than small boxes. As shown in Fig. 5(a), the number of positive samples assigned for larger sheep is more than smaller sheep. Therefore, the model focuses too much on large ground truth boxes, neglecting small ground truth boxes. It decreases the detection performance.
- There is an inconsistency between the IoU-based method and network optimization. Detectors optimize classification and localization objectives simultaneously, while the IoU-based method only uses localization quality (IoU score) to perform the assignment. It leads to insufficient information when selecting positive samples.
- Separating positive/negative samples lacks context, and it leads to improper detection.
- The detection performance is sensitive to threshold $t_p$ and $t_n$.

The above problems cause sub-optimal assigning results in IoU-based methods and open research directions for improving anchor assignment.

**Table 1**
The comparison of hard anchor assignment and soft anchor assignment between different object detectors in theoretical analysis and performance on MS-COCO dataset.

| Method | Prior Anchor | Positive Bag Construction | Cost metric | Re-assignment | Addit. |
|---|---|---|---|---|---|
| | | | **Hard Anchor Assignment** | | |
| RetinaNet [107] | Anchor box | $IoU(a_i, g_j) \geqslant 0.5$ | spatial IoU, scale | – | – |
| FCOS [118] | Anchor point | Inside GT boxes (center prior) | spatial center, scale | – | $r = 1.5$ |
| ATSS [140] | Anchor box, anchor point | Top-$k$ = 9 anchor boxes, its centers are closest to ground truth center | spatial IoU, scale | Assuming IoU scores as Gaussian distribution. Setting new assignment threshold: mean + standard deviation. | – |
| YOLOF [124] | Anchor box | Top-$k$ = 4 nearest anchor boxes | $L1$ localization cost | – | – |
| | | | **Soft Anchor Assignment** | | |
| FreeAnchor [148] | Anchor box | Top-$k$ IoU anchor boxes | classification and localization loss | Maximizing detection customized likelihood. (learn to match positive and negative anchors). | – |
| NoisyAnchor [149] | Anchor box | Top-$k$ IoU anchor boxes | cleanliness score | Using cleanliness scores as soft labels. | – |
| MAL [150] | Anchor box | Top-$k$ IoU anchor boxes | classification and localization loss | All-to-Top-1 selection strategy during learning. | – |
| AutoAssign [152] | Center weighting | Inside GT boxes (center prior) | classification and localization loss | Confidence weighting reshapes the negative and positive weighting. | – |
| PAA [151] | Anchor box | $IoU > 0.1$ | classification and localization loss | Computing anchor scores. Selecting top-$k$ = 9 smallest scores. Finding two Gaussian via Fitting GMM to top-$k$ scores. Middle of positive anchor distribution as the separation boundary. | $\lambda = 1.3$ |
| DETR [156] | Object query | All queries | classification and localization cost | One-to-one assignment using Hungarian algorithm. | $\lambda_{L1} = 5$ $\lambda_{IoU} = 2$ |
| POTO [153] | Anchor point | Inside GT boxes (center prior) | classification and localization score | One-to-one assignment using Hungarian algorithm. Applying auxiliary one-to-many assignment of ATSS. (For computing auxiliary loss). | $\alpha = 0.8$ |
| LLA [162] | Anchor box, Anchor point | All anchor boxes All anchor points | classification and localization cost | Selecting top-$k$ = 10 minimum cost. | $\lambda = 1.5$ $C = 10^2$ |
| OTA [154] | Anchor box, Anchor point | Top-$r$ = 5 anchors whose centers are closest to object center | classification and localization cost, center prior | Computing optimal assigning plan via Sinkhorn-Knopp Iteration. Dynamic $k$ estimation. | $\alpha = 1.5$ |
| YOLOX [124] | Anchor point | Inside GT boxes (center prior) | classification and localization cost | Selecting top-$k$ minimum cost. Dynamic $k$ estimation. | $r = 2.5$ $\lambda = 3.0$ |
| Musu [163] | Anchor point | Quality scores $\geqslant t$ | classification and localization score | Mutually learning with the sampling method | $\theta = 4.0$ $b = 0.1$ |

FCOS [118] considers anchor box $a_i$ as anchor point. The anchor point $a_i$ is assigned as a positive sample if it satisfies two conditions: spatial constraint and scale constraint. Specifically, if anchor point $a_i$: (1) falls into the center region of ground truth box $g_j$ and (2) belongs to the regression range defined for each pyramid level, this point is assigned as a positive sample. Anchor points that do not satisfy two conditions are labeled as negative samples. Fig. 5 (b) shows how to define positive and negative samples in FCOS based on spatial and scale constraints. As a result, FCOS produces many positive samples for each object and thus brings sufficient information of ground truth boxes to efficiently train the bounding box regressor. Although FCOS achieves better performance than RetinaNet, it does not take all problems of hard assignment into account.

ATSS [140] proposes adaptive training sample selection that automatically assigns positive and negative samples based on a statistical distribution of the object. For a ground truth box $g_j$, the procedure is briefly computed as:

1. Compute $IoU(a_i, g_j)$ for all anchors.
2. Compute the center distance between all anchors and the ground truth using L2 distance.
3. On each pyramid level, select $k$ anchor boxes whose centers are closest to the ground truth center.
4. Get corresponding IoU for these anchor candidates, and compute the mean and standard deviation.
5. IoU threshold $t_p$ is the sum of the mean and standard deviation.
6. Assign $a_i$ as positive if $IoU(a_i, g_j) \geqslant t_p$.

Adaptive training sample selection is demonstrated effective and has been used by many recent state-of-the-art one-stage detectors such as [141–146]. Although ATSS overcomes the limitation of the hard IoU threshold and improves the detection performance, it still lacks classification cost and feature context. And computing mean and standard deviation based on normal distribution do not satisfy the arbitrary distribution of real datasets [147].

YOLOF [124] introduces Uniform Matching to solve scale imbalance on positive samples originated by large ground truth bounding boxes. Uniform Matching is described as:

1. Compute the $L1$ cost between anchor boxes and ground truth boxes.
2. Select $k$ *nearest* anchors as positive samples for each ground truth box according to $L1$ cost.
3. Ignore large $IoU > 0.7$ negative anchors and small $IoU < 0.15$ positive anchors.

This strategy makes the number of positive samples uniform for all ground truth boxes. Hence, small and large ground truth boxes all join during training and contribute equally. However, Uniform Matching only considers the localization cost. It still produces noisy and ambiguous anchors due to lack of classification cost and model's feedback, and hinders the optimization.

**Table 2**
Comparative performance between existing object detectors on the benchmark MS-COCO dataset.

| Method | Backbone | Schedule | $AP$ | $AP^{50}$ | $AP^{75}$ | $AP^{S}$ | $AP^{M}$ | $AP^{L}$ |
|---|---|---|---|---|---|---|---|---|
| **Hard Anchor Assignment** | | | | | | | | |
| RetinaNet [107] | ResNet-50-FPN | 1× | 36.5 | 55.4 | 39.1 | 20.4 | 40.3 | 48.1 |
| FCOS [118] | ResNet-50-FPN | 1× | 38.7 | 57.4 | 41.8 | 22.9 | 42.5 | 50.1 |
| ATSS [140] | ResNet-50-FPN | 1× | 39.4 | 57.6 | 42.8 | 23.6 | 42.9 | 50.3 |
| YOLOF [124] | ResNet-50-DE | 1× | 37.5 | 57.0 | 40.4 | 19.0 | 42.0 | 53.2 |
| **Soft Anchor Assignment** | | | | | | | | |
| FreeAnchor [148] | ResNet-50-FPN | 1× | 38.7 | 57.3 | 41.5 | 21.0 | 42.0 | 51.3 |
| NoisyAnchor [149] | ResNet-50-FPN | 1× | 38.0 | 56.9 | 40.6 | – | – | – |
| MAL [150] | ResNet-50-FPN | 1× | 39.2 | 58.0 | 42.3 | – | – | – |
| AutoAssign [152] | ResNet-50-FPN | 1× | 40.4 | 59.6 | 43.7 | 22.7 | 44.1 | 52.9 |
| PAA [151] | ResNet-50-FPN | 1× | 40.4 | 58.4 | 43.9 | 22.9 | 44.3 | 54.0 |
| DETR [156] | ResNet-50-FPN | 12× | 40.1 | 60.6 | 42.0 | 18.3 | 43.3 | 59.5 |
| POTO [153] | ResNet-50-FPN | 3× | 41.4 | 60.1 | 44.9 | 25.6 | 44.9 | 53.1 |
| OTA [154] | ResNet-50-FPN | 1× | 40.7 | 58.6 | 44.1 | – | – | – |
| Musu [163] | ResNet-50-FPN | 1× | 40.6 | 58.9 | 44.3 | 23.0 | 44.0 | 54.2 |

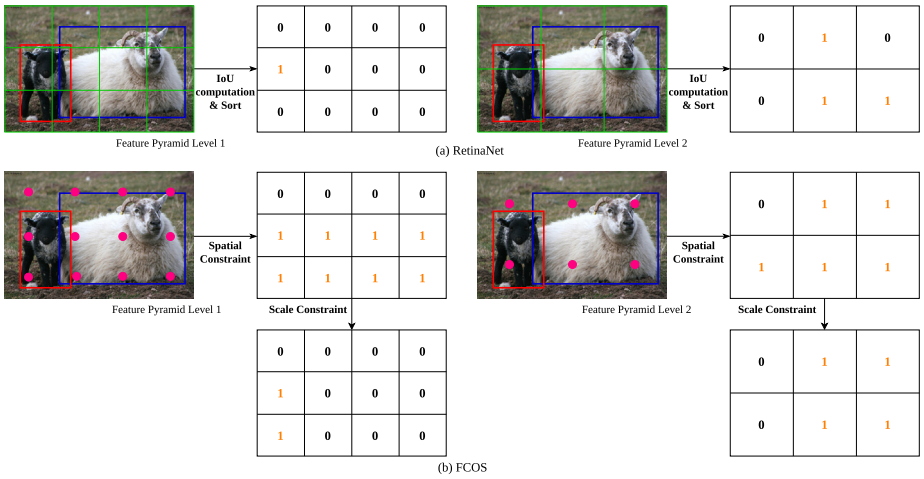



**Fig. 5.** (a) Retinanet [107] uses IoU-based anchor assignment to separate anchor boxes from two feature pyramid levels into positive and negative samples. (b) FCOS [118] utilizes two constraints (spatial and scale constraints) to divide anchor points from pyramid level 1 and 2. Blue boxes denote anchor boxes, and pink points indicate anchor points. Red and blue boxes denote ground truth bounding boxes. Positive samples are labeled as 1. Negative samples are labeled as value 0.

### 3.2. Soft assignment methods

In recent years, state-of-the-art detectors tried to improve anchor assignment to be more adaptive instead of hard IoU threshold and generate the best assignment results according to detection prediction and additional network optimization. These detectors take noisy anchors and ambiguous anchors into consideration, and the detection model needs to participate in anchor assignment. Fig. 6 illustrates the general computation of soft anchor assignment methods.

FreeAnchor [148] trains the detector using Maximum Likelihood Estimation (MLE), where detection customized likelihood is proposed to unify the classification and localization. Positive samples are determined by maximizing the likelihood via updating the IoU-based method and learning-to-match method, as follows:

1. For each ground truth, an anchor bag is constructed by selecting *topK* anchor boxes based on IoU between anchor boxes and the ground truth.
2. Maximizing detection customized likelihood corresponds to minimizing anchor matching loss, i.e., learn to match positive and negative anchors from anchor bag, to find the suitable positive anchors automatically.

However, the learning-to-match method according to MLE is the non-convex objective function. Thus, it leads to hard optimization and sub-optimal problem.

NoisyAnchors [149] explores the model's feedback to propose cleanliness scores for anchor boxes. The cleanliness score of an anchor is computed as a linear combination of localization quality from localization branch and classification score from classification branch, as follows:

$$c = \begin{cases} \alpha \times IoU(a_i^r, g_j) + (1-\alpha) \times c^{cls} & for\, a_i \in \mathscr{P} \\ 0 & for\, a_i \in \mathscr{N}, \end{cases} \quad (2)$$

where $\alpha$ is a factor to balance the classification score $c^{cls}$ and localization quality-$IoU(a_i^r, g_j)$ between regressed anchor $a_i^r$ and ground truth $g_j$. The cleanliness scores $c$ are used as soft labels in classification loss for adjusting the contribution of different anchors to this loss, and sample re-weighting factors in both classification and localization losses to down-weight the contribution of noisy anchors and make the model focus on clean anchors. The positive sample set $\mathscr{P}$ before joining with the network is assigned by sorting IoU scores between anchor set and ground truth set and then choosing $K$ anchors with the highest ranking as positive samples.

Similar to [148,149], Multiple Anchor Learning (MAL) [150] firstly constructs positive candidates for each object based on top$k$ IoUs between anchor boxes and a ground truth box, and then selects proper positive boxes from candidate boxes. The constructed boxes are forwarded to the network to output classification and localization confidences. However, updating the network parameters through SGD is a difficult problem and can lead to suboptimal results. Therefore, MAL proposes anchor

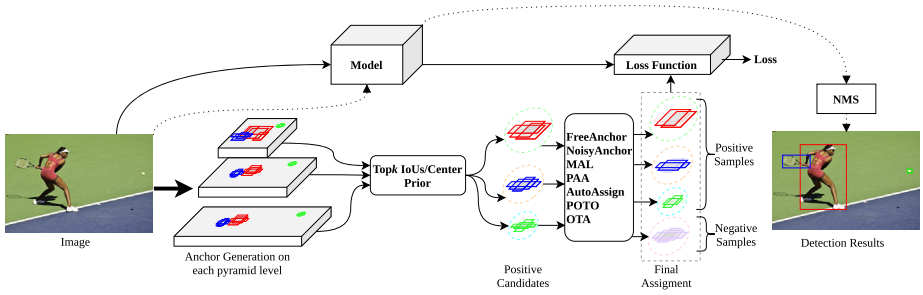



**Fig. 6.** The general pipeline of soft anchor assignment consists of two procedures: (1) construct positive candidate boxes according to top*k* IoUs in anchor-based methods, or center prior (center sampling) in anchor-free methods; (2) select final assignment by using FreeAnchor [148], NoisyAnchor [149], MAL [150], PAA [151], AutoAssign [152], POTO [153], and OTA [154]. Positive samples are addressed by visible red, blue, and green boxes. Other boxes are negative samples. Solid lines indicate training flow, and dash lines denote testing flow.

depression that reduces the confidence of selected anchors by perturbing its features via attention maps. To select final positive samples from the anchor bag, MAL introduces the All-to-Top-1 strategy that linearly reduces the number of positive samples from all boxes (first epoch) to 1 box (last epoch) during training.

Inspired by center prior knowledge (spatial constraints) in FCOS [118], AutoAssign [152] introduces the differentiable anchor assignment that automatically separates anchor boxes into positive and negative samples in a data-driven approach based on Center Weighting and Confidence Weighting. Firstly, Center Weighting models each category distribution using Gaussian distribution with location offsets inside a ground truth box as variables. Mean and standard deviation are optimized via the backward pass. Secondly, Confidence Weighting reshapes the positive and negative weightings of the ground truth locations in both spatial constraint and scale constraint. Finally, the loss values of positive and negative samples will be computed and optimal anchor assignment will be executed jointly with the detection network.

PAA [151] adaptively separates a set of anchors into positive/negative samples for a ground truth bounding box in a probabilistic manner. Firstly, PAA defines an anchor score that reflects the quality of the detected bounding box. This score is computed by multiplying the classification score and IoU score. To train the model, anchor scores are converted to classification and localization losses by using the negative logarithm. Secondly, with these computed anchor scores, PAA applies Gaussian Mixture Model (GMM) of two modalities (i.e., corresponding to two sets: positive sample set and negative sample set) conditioned on the model's parameters to represent the distribution of the anchor scores. Based on anchor probabilities, the boundary of two sets is identified. For each ground truth box $g_j$, the PAA procedure is summarized as:

- Get all anchors that overlap with ground truth box $g_j$ based on IoU scores.
- For each feature pyramid, compute anchor scores between selected anchors and the ground truth box. Then, choose the *topK* smallest scores.
- Fit GMM to *topK* smallest scores to find probabilities of two Gaussian.
- Use the middle of positive anchor distribution as the separation boundary.Although PAA considers the model's feedback (classification and localization losses) into account, it still relies on hard thresholds, distribution assumptions, and other prior knowledge for performing the assignment.

According to PAA, Label Assignment Distillation (LAD) [155] adopts the knowledge distillation technique that uses a small teacher network to produce training samples for a student network. This method is a new perspective in anchor assignment literature, which complements existing soft assignment approaches.

DETR [156] presents the new end-to-end object detection method, eliminating hand-crafted designs such as Non-Maximum Suppression (NMS), anchor generator. To remove NMS post-processing, DETR introduces bipartite matching (i.e., optimal global matching) that computes one-to-one matching between predictions (i.e., object queries) and ground truth box according to the global cost matrix *C*. More specifically, object queries are a set of learnable positional encoding to reason about the correlation between objects and image global features to yield the object's coordinates and classes, viewed as a set of predictions that initially encodes the information of objects about positional and semantic features. In practical implementation, object queries are learned from random initialization [156], attached anchor points [157], and attached anchor boxes [158]. Each element of the matrix *C* are a weighted sum of three components, as follows:

$$c_{ij} = c_{ij}^{cls} + c_{ij}^{loc} = \hat{p}_{\sigma(i)}(l_i) + \lambda_{iou}\mathscr{L}_{iou}(b_{\sigma(i)}, g_j) + \lambda_{L1}\left\|b_{\sigma(i)} - g_j\right\|_1, \quad (3)$$

where $\hat{p}_{\sigma(i)}$ is class probability of class $l_i$. $b_{\sigma(i)}$ is the predicted bounding box. The localization cost is a linear combination of the $L1$ loss and GIoU loss ($\mathscr{L}_{iou}$) [159], where $\lambda_{iou}, \lambda_{L1}$ are the balancing coefficients. After computing the matching cost, the Hungarian algorithm is used to find the optimal bipartite matching. As a result, one ground truth box is only assigned with one prediction without duplicates. Fig. 7 shows the difference between one-to-one assignment and one-to-many assignment in detection literature and the benefit of one-to-one prediction in term of simple network and efficient design.

POTO [153] proposes a prediction-aware one-to-one anchor assignment, which dynamically determines positive samples for each ground truth box. Instead of matching cost in DETR [156] and its application [160], POTO defines the matching quality of classification and localization estimated from model learning status. Given the *N* predictions and *G* ground truth boxes, the optimal matching is computed as:

$$\hat{\sigma} = \arg\max_{\sigma \in \mathfrak{S}_G^N} \sum_{i}^{|G|} Q_{i,\sigma(i)}, \quad (4)$$

where $\mathfrak{S}_G^N$ is a permutation of N elements. $Q_{i,\sigma(i)}$ is the matching quality that takes spatial prior, classification score, and localization quality into account. It is defined as:

$$Q_{i,\sigma(i)} = \mathbb{1}[\sigma(\mathbb{i}) \in \mathbb{A}_\mathbb{i}] \cdot \left(\hat{\mathbb{p}}_{\sigma(\mathbb{i})}(\mathbb{l}_\mathbb{i})\right)^{1-\alpha} \cdot \left(\mathbb{I}_\mathbb{o}\mathbb{U}(\mathbb{b}_\mathbb{i}, \mathbb{g}_\mathbb{i})\right)^{\alpha} \quad (5)$$

The spatial prior is denoted by $\mathbb{1}[\sigma(\mathbb{i}) \in \mathbb{A}_\mathbb{i}]$, which selects anchor candidates $A_i$ for $i^{th}$ ground truth based on the spatial constraint in FCOS [118] (e.g., if an anchor falls into the center region of ground

truth box, this anchor is considered as one candidate). The classification score $(\hat{p}_{\sigma(i)}(l_i))$ and localization quality $(IoU(b_i, g_i))$ are multiplied to present the matching quality for ground truth $g_i$ where $b_i$ is the predicted box from anchor box/point $a_i$. To find the optimal one-to-one matching $\hat{\sigma}$, the Hungarian algorithm is applied for maximizing the matching quality. To make the anchor assignment more effective, POTO uses an auxiliary loss according to one-to-many assignment in ATSS [140] to enhance the strong and robust feature representation. Both DETR and POTO use one-to-one matching, which removes the hard NMS and becomes end-to-end detectors. However, POTO still relies on hand-crafted design for selecting anchor candidates.

OneNet [161] demonstrates the one-to-one assignment is the key factor to achieve end-to-end object detection and analyzes the essential components in solving one-to-one matching. OneNet states that the classification cost is the main component along with localization cost and can reduce noisy anchors originated by localization cost.

Unlike existing methods [149,151,153] which only compute the matching costs between anchor candidates (based on IoU score, localization cost) and its assigned ground truth box, LLA [162] computes the matching costs between all anchors and ground truth boxes, as follows:

$$C = C^{cls} + \lambda C^{loc} + C^{inbox} \tag{6}$$

$$= \mathscr{L}_{cls}(L, P(\theta, A)) + \lambda \mathscr{L}_{loc}(G, B(\theta, A)) + C^{inbox},$$

where the cost matrix $C \in \mathbb{R}^{|G| \times N}$ is a linear combination of classification cost $C^{cls} \in \mathbb{R}^{|G| \times N}$ and localization cost $C^{loc} \in \mathbb{R}^{|G| \times N}$, where $N$ and $|G|$ are number of anchor boxes/points and number of ground truth boxes, respectively. $\lambda$ is balancing the range of two costs. $P(\theta, A) \in \mathbb{R}^{|A| \times L}, B(\theta, A) \in \mathbb{R}^{|A| \times N}$ are the score predictions and box predictions, where $\theta$ is parameters of the detection model. $C^{inbox}$ is spatial prior that defined in POTO [153]. This spatial prior is added to the cost matrix to control the model converge. Instead of one ground truth assigned to one anchor, LLA assigns multiple positive samples for one ground truth by selecting *topK* smallest values on each row of cost matrix $C$ as positive samples and others as negative samples. If one anchor is assigned to multiple ground truth boxes, the anchor-ground truth pair with the smallest cost is selected. Therefore, LLA performs anchor assignment in a fully adaptive manner based on the model's feedback, which can solve the crowd occlusion problem in pedestrian detection.

OTA [154] formulates anchor assignment into an optimal transport problem. This method considers one ground truth $g_i$ as one supplier and one anchor as one demander. The cost $c_{ij}$ to transport one positive unit from $g_i$ to anchor $a_j$ is computed as a linear combination of classification and localization costs that are similar to previous methods [156,162]. For negative samples, the background class is another supplier. The cost of this background and an anchor is only calculated as the pair-wise classification cost. The solution of optimal transport plan corresponding to the global assignment results is solved through off-the-shelf Sinkhorn-Knopp Iteration. An anchor box is assigned as a positive sample if this anchor box receives enough information of positive label from a ground truth box $g_i$.

Optimizing the OTA objective through Sinkhorn-Knopp Iteration is hard optimization and takes much additional time for converging the detection model. To overcome this problem, YOLOX [164] simplifies the OTA assignment by utilizing a dynamic top-$k$ operation. This assignment is called SimOTA. Instead of selecting positive anchors via Sinkhorn-Knopp, SimOTA chooses the top-$k$ smallest costs corresponding to $k$ positive anchors for a ground truth box.

Feng et al. [165] points out that hard anchor assignment methods such as RetinaNet [107], FCOS [118], and ATSS [140] generate assignment results that a spatial location of a positive sample does not contain the center of the object, and thus, the detection model produces mislocalized and misclassified predictions. From these observations, TOOD proposes Task-aligned Sample Assignment that includes two new designs: a sample assignment procedure and a task-aligned loss. As shown in Fig. 6, this method has the same computation as the previous soft anchor assignment. Firstly, TOOD constructs a positive anchor bag by selecting top$k$ detection quality where detection quality for one anchor is defined as the multiplication of classification score and IoU score between the predicted box and the ground truth. Secondly, the detection quality is used for reweighting localization loss and representing the classification target.

Instead of using the same anchor assignment procedure, Musu [163] assigns different training samples for classification and localization tasks and learns task relatedness through mutual supervision. Like common soft anchor assignments, Musu constructs positive candidates for each object. Given ground truth box $g_j$, model's feedback $\{p_i, IoU(a_i, g_j)\}$, the quality score $P_i$ used as a cost metric is defined as:

$$P_i = p_i IoU(a_i, g_j)^{\theta}, \tag{7}$$

where $\theta$ is a scaling factor. To adaptively select positive samples for ground truth box $g_j$, Musu computes the threshold $t$ as follows:

$$t = b \cdot \max_i P_i, \tag{8}$$

where $b$ is a controlling factor. If $P_i \geqslant t$, this anchor is assigned as a positive sample. To mutually supervise the classification and local-

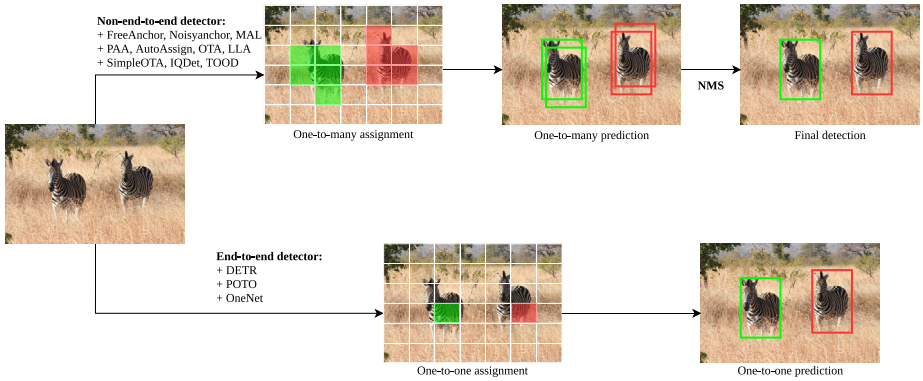



**Fig. 7.** The definition of non-end-to-end detection and end-to-end detection relies on the definition of soft anchor assignment: one-to-many assignment and one-to-one assignment. The non-end-to-end detector requires the hand-crafted NMS procedure to remove duplicate bounding boxes, and this post-processing takes additional latency.

ization head, this method reweights the importance of each assigned sample based on the ranking of quality scores.

## 4. Sampling methods

When training the object detection model, most anchor boxes are assigned as negative samples. Using all samples with the equal contribution for updating the detection model leads to extreme imbalance. This imbalance causes: (1) training detector is ineffective since negative samples do not contain useful features for localizing object coordinates, (2) the number of negative samples is too large for one object, and it can overwhelm training and hamper the detection performance. To verify our theoretical analysis, we illustrate the imbalance between positive and negative samples on the MS-COCO [95] dataset in Table 3. As a result, in the popular detector RetinaNet [107] and FCOS [118], the number of positive samples is much smaller than the number of negative samples. Therefore, when not figured out, the model heavily pays attention to negative samples, and this problem directly degrades the final detection performance.

The solution of this problem is grouped into two methods: hard sampling methods and soft sampling methods. Fig. 8 illustrates the difference between hard sampling methods and soft sampling methods during training.

### 4.1. Hard sampling methods

From labeled samples in the anchor assignment procedure, hard sampling methods select the useful set of positive samples and negative samples, and discard a certain amount of non-useful samples during training. Therefore, selected samples uniformly contribute to the detection loss, and non-selected samples do not contribute to the classification and regression losses. Table 2 illustrates the strategy of the hard sampling methods in training detection model.

Random sampling has been widely used in many conventional detectors such as two-stage detectors [64,73,74,108,125], randomly selecting an amount of samples based on a pre-defined ratio from two sets. In the RPN stage, the ratio is set to $1:1$ of 256 sampled anchors corresponding to 128 positive samples and 128 negative samples from an image in a mini-batch. If the number of selected positive samples is less than fixed values, it is padded with random negative samples. For training the R-CNN network, 16 positive RoIs and 48 negative RoIs are randomly selected in each mini-batch.

Instead of equally giving the contribution of selected negative samples to the detection loss, SSD [97] states that training detectors on hard negative samples achieve faster optimization (bring informative gradients for computing back-propagation), more stable training, and better performance. The method selects negative samples with high losses, called hard-sample mining. Firstly, the model with initial learnable parameters is trained on a subset of random negative samples. Secondly, this method picks the false positives (hard samples) and train a new classifier on them again. These steps are executed iteratively until the model converges. OHEM [91] proposes online hard example mining, which samples both negative and positive samples based on loss criterion. However, OHEM method leads to extra memory, more training times, and noisy samples.

Rather than selecting negative samples according to their loss values, Libra R-CNN [90] introduces an IoU-balanced sampling method that selects negative samples based on IoU intervals. Firstly, this method computes the IoU scores between negative samples and ground truth boxes. Secondly, IoU intervals are split into $K$ bins, and then candidate negative samples are uniformly distributed to each bin. Finally, negative samples with higher losses are selected equally from each bin.

**Table 3**
The statistics of the number of positive and negative samples

| Method | #positive samples | #negative samples | #total samples |
|---|---|---|---|
| RetinaNet [107] | 162 | 167687 | 167849 |
| FCOS [118] | 209 | 19743 | 19952 |

Inspired by the IoU-balanced sampling, Overlap Sampler [166] computes overlaps among samples (i.e., compute the IoU scores between positive and negative samples) instead of computing with ground truth boxes. And then, this method uses overlap scores to implement sampling that is the same procedure as IoU-balanced sampling.

### 4.2. Soft sampling methods

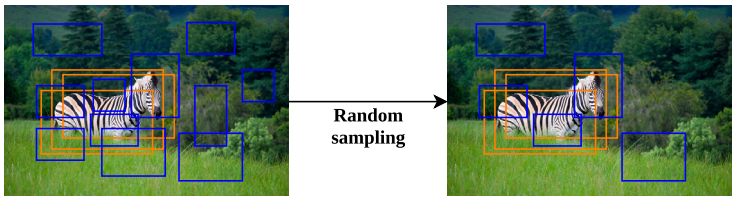
Instead of selecting a sub-set from positive and negative samples, soft sampling methods use all assigned samples during training by controlling the contribution of each sample to the classification loss according to the usefulness of each positive and negative sample. Each sample $i$ is attached with a weighting factor $\omega_i$ measured by classification scores, IoU scores, or both to reshape the loss: $\omega_i \mathscr{L}_{cls}$. Table 4 shows the comparative illustration of soft sampling methods. In this Table, hard samples indicate misleading samples that produce high losses during training. Because of the uncertainty problem (such as complex scenes, occlusion, ambiguities, blur, and shadow), the anchor assignment procedure yields ambiguous samples, for example, the samples belong to the boundary of the positive and negative sets, such as anchor boxes are misleadingly assigned as positive or negative due to ambiguities or occlusion. For example, Fig. 9 shows three hard samples that were misassigned as positive samples. In the first image, due to occlusion, the green box labeled as a positive sample for motorcycle class is a hard sample since this green box contains semantic information of the person class. Thus, this green box is not only a hard sample for motorcycle class but also for the person class and it leads to ambiguous learning of the detection models. In the second image, most of the pixels in the blue box belong to the background class and thus, this box is a hard samples for the person class. This problem directly decreases the detection accuracy.

Focal loss [107] was the first method that dynamically reshapes Cross-Entropy $CE$ loss, defined as:

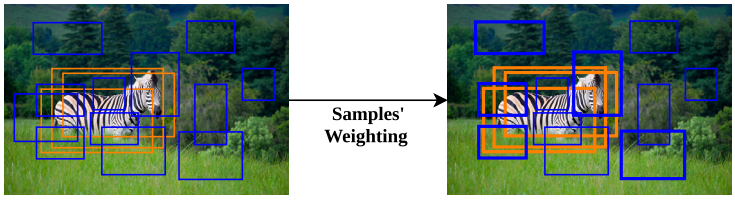
$$FL(p_i) = \omega_i CE(p_i) = -\alpha(1-p_i)^{\gamma} CE(p_i), \tag{9}$$

where $\alpha$ is a balanced variant factor, and $p_i$ is the classification score of a sample $i$ for the ground truth class. If the value $p_i$ is close to 1 (well-classified sample corresponding to easy sample), the weighting factor $\omega_i$ approaches to 0, and the loss for this sample is down-weighted. When a hard sample has a low confidence score (e.g., misclassified sample) and the $\omega_i$ goes to 1, the loss is larger. Thus, FL loss promotes hard samples and down-weights easy samples.

GFL [141] and VFNet [143] extend the Focal loss to a new continuous version joining classification score and localization quality as new label. Similar to Focal loss, GFL proposes a weighting factor: $\omega_i = |q_i - p_i|^{\beta}$ where $q_i$ is continuous target score for sample $i$ and $p_i$ is prediction score. VFNet computes a weighting factor for two cases: (1) $\omega_i = q_i$ when $q > 0$; (2) $\omega_i = \alpha q_i^{\gamma}$ when $q = 0$. Case (2) will down-weight the contribution of negative samples and case (1) will not affect the loss values of positive samples. Therefore, VFNet's strategy focuses on the contribution of high-quality positive samples to classification loss than hard samples.




(a) Hard sampling methods




(b) Soft sampling methods

**Fig. 8.** The difference between (a) hard sampling methods and (b) soft sampling methods. Orange boxes indicate positive samples, and blue boxes denote negative samples. Hard sampling methods select a subset of assigned anchors by randomly removing some negative samples based on a pre-defined ratio during training. Otherwise, all samples in soft sampling methods are joined during training and automatically assigned by weighting based on their usefulness. For example, thicker boxes address greater weighting values.

**Table 4**
The comparison of hard sampling methods and soft sampling methods between different detectors in the aspects: criterion, strategy, and performance on MS-COCO dataset.

| Method | Criterion | Strategy | $AP$ | $AP^{50}$ | $AP^{75}$ |
|---|---|---|---|---|---|
| | | **Hard Sampling Methods** | | | |
| Random Sampling [74] | pre-defined ratio | randomly selecting an amount of positive and negative samples. | 37.2 | 59.3 | 40.3 |
| Hard sample mining[97] | classification loss | - selecting negative samples with high losses.<br>- training a new classifier. | 29.5 | 49.3 | 30.9 |
| OHEM [91] | classification loss | selecting positive and negative samples with high losses. | 37.4 | 59.5 | 40.3 |
| IoU-based sampling[90] | $IoU(a_i, g_j)$, loss | - distributing equal number of negative samples to each IoU bin.<br>- equally selecting negative samples with higher losses from each bin. | 38.3 | 59.5 | 41.9 |
| Overlap sampler [166] | $IoU(a_i^{pos}, a_i^{neg})$ | - selecting negative samples with high IoU scores.<br>- focusing on easy positive samples with high IoU scores based on ranking-based method and sample weighting. | 38.6 | 60.2 | 41.9 |
| | | **Soft Sampling Methods** | | | |
| Focal loss [107] | classification score | - focusing on hard samples with high losses.<br>- down-weighting the contribution of easy samples. | 36.5 | 55.4 | 39.1 |
| GFL [141] | classification score, IoU score | - focusing on hard samples with high losses.<br>- down-weighting the contribution of easy samples. | 40.2 | 58.4 | 43.3 |
| VFNet [143] | classification score, IoU score | - focusing on easy samples with high IoU scores.<br>- down-weighting the contribution of hard samples. | 41.6 | 59.5 | 45.0 |
| Sampling-Free [167] | classification loss, localization loss, sample uncertainty | dynamically adjusting classification loss based on guided factor. | 38.4 | 59.9 | 41.7 |
| GHM [168] | classifcation score | focusing on hard samples by down-weighting the contribution of easy samples and outliers. | 37.0 | 55.5 | 39.2 |
| SWN [169] | classification score, IoU score, classification loss, localization loss | - focusing on easy samples.<br>- down-weighting the contribution of hard samples with high uncertainty. | 38.5 | 58.7 | 42.1 |
| PISA [94] | classification score, IoU score | - focusing on positive samples with higher IoU scores.<br>- focusing on negative samples with higher classification scores. | 38.8 | 59.3 | 42.7 |
| DW [170] | classification score, IoU score | - focusing on positive samples with higher IoU and classification scores.<br>- focusing on negative samples with lower IoU scores. | 41.5 | 59.8 | 44.8 |

Sampling-Free [167] analyzes the performance of training detector with sampling strategy (using Focal loss) and without sampling (using Cross-Entropy (CE) loss). The experimental results show that RetinaNet with CE has undesirable stability and leads to inappropriate classification gradient magnitude due to bias initialization and loss weighting. Instead of weighting the samples, the Sampling-Free method dynamically adjusts CE loss by introducing guided factor: $\frac{g^t}{\sigma^2}$, where $g^t = \frac{\mathscr{L}_{loc}}{\mathscr{L}_{CE}}$ and $\sigma$ is uncertainty weighting [171].

Through empirical experiments, GHM [168] observed that there is an imbalance in gradient norm distribution: (1) Too many easy negative samples have slight gradient and overwhelm the contribution of other samples, (2) the number of samples with large gradient norm is bigger than the number of samples with medium gradient norm. From this analysis, GHM introduces a gradient harmonizing mechanism that balances gradient norm distribution of easy and hard samples, computed as: $\omega_i = \frac{N}{GD(a_i)}$, where $N$ is the number of total samples in a mini-batch, $GD(a_i)$ is the fraction number of samples that have similar gradient norm with sample $a_i$.

SWN [169] views the sampling procedure in a probabilistic perspective and computes the sample's weighting through uncertainty prediction for both classification and localization tasks.

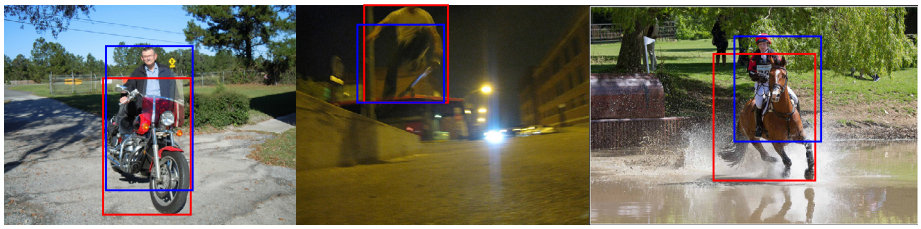


**Fig. 9.** Some hard samples of the MS-COCO dataset. Red boxes indicate ground truth bounding boxes, and blue boxes denote hard samples.

The sample weightings are defined as: $\omega_i^{cls} = e^{-2 * m_i^{cls}}$ for classification loss and $\omega_i^{loc} = e^{-2 * m_i^{loc}}$ for localization loss, where $m_i^{cls}$ and $m_i^{loc}$ are learned via several stacked fully-connected layers in which current loss, classification score and IoU score are concatenated as input feature. SWN gives the weights to useful samples by down-weighting the contribution of uncertainty samples (hard samples) and focusing the model on certainty samples (easy samples). Moreover, these weighting factors are used to leverage multi-task learning in detection learning. However, this goal is beyond the scope of this paper.

PISA [94] analyzes the importance of positive samples and negative samples with regards to AP metric: (1) Based on IoU score between positive samples and its assigned ground truth box, the sample with higher IoU score is more important, (2) Based on foreground classification scores, the negative sample with the larger score is more important. From these observations, PISA proposes IoU-Hierarchical Local Rank (IoU-HLR) and Score-Hierarchical Local Rank (Score-HLR) to sort the importance of positive samples and negative samples, respectively. Specifically, IoU-HLR ranks positive samples as follows:

1. Compute IoU between predicted bounding boxes and the corresponding ground truth.
2. Split all positive samples into various groups.
3. Sort the samples within each group based on IoU scores with descending order.
4. Sort again within same-rank group.

Similarly, for negative samples, Score-HLR is performed as:

1. Take the maximum positive score prediction over all foreground classes of each negative sample as $s_i$.
2. Suppress negative samples whose $s_i \leqslant t_n$, the left samples are valid samples.
3. Divide valid samples into different groups using NMS-Match.
4. Rank the matched samples in two steps to get Score-HLR: (1) In the same group, rank samples with their scores; (2) In the same score rank across different groups, rank samples with their scores again.

Finally, PISA linearly maps both IoU-HLR and Score-HLR to the final weighting factors, computed as:

$$\omega_i = ((1-\beta)u_i + \beta)^{\gamma}, \tag{10}$$

where $u_i = \frac{N_{max} - r_i}{N_{max}}$ is a normalized rank for sample $i$ in which $N_{max}$ is a maximum value of samples over all classes, $r_i$ is the rank of sample $i$. $\beta$ controls the importance of normalized rank. $\gamma$ is a modulating factor. Following common strategies, the weighting factor $\omega_i$ is attached to the classification loss function, giving more contribution of positive samples with high IoUs and negative samples with high scores to this loss.

To better learn the correlation and consistency among tasks, DW [170] separately computes sample weightings of the negative and positive samples as follows:

$$\omega_i^{pos} = e^{\mu p_i IoU^{\beta}} \times p_i IoU^{\beta}, \tag{11}$$

$$\omega_i^{neg} = \begin{cases} p_i^{\gamma_2}, & \text{if } IoU < 0.5 \\ (-k \times IoU^{\gamma_1} + b) \times p^{\gamma_2}, & \text{if } IoU \in [0.5, 0.95] \\ 0, & \text{if } > 0.95, \end{cases} \tag{12}$$

where $\mu, \beta, \gamma_1, \gamma_2$ are hyper-parameters to balance prediction values. As seen in the equations, DW method upgrades the contribution of positive samples with high IoU and classification scores. And for negative samples, DW method focuses on hard samples with low IoU scores.

RS Loss [172] proposes the new sample weighting according to the ranking method: (1) RS Loss ranks each positive sample that is higher than all negative samples, (2) RS Loss sorts positive samples with respect to localization IoU scores. Differently, EQL v2 [173] observes that there is a gradient imbalance between positive and negative samples in long-tailed object detectors and EQL [174]. Based on this insight, EQL v2 increases the gradient of positive samples and decreases the gradient of negative samples by computing the ratio of accumulated gradients.

## 5. Recent trends in object detection

Convolutional Neural Networks (CNNs) such as [119,120,175–180] have dominated the field of computer vision, proving the generalization capability in both modeling and learning. Because the receptive field of convolution operation is limited to the local regions, previous methods [181–183,89,127,184–186] design channel and spatial attention mechanisms to model long-range dependencies in the visual inputs that complement local convolution operation. These methods have demonstrated the effectiveness on different computer vision tasks such as image classification, object detection, semantic/instance segmentation. Inspired by the success of the attention mechanism in visual tasks, in recent years, many researchers have adapted and facilitated Transformer [187] architecture to object detection, which achieves significant improvements in both global computation and performance, and establishes new state-of-the-art detectors on the challenging benchmark [95]. Transformer architecture originally was designed for a sequence-to-sequence machine translation, which becomes the de facto standard method in most natural language processing. The core element of the Transformer is the self-attention block that models long-range dependencies in data. This promising property brings many advantages to solving visual tasks such as general modeling capacity (relation of pixel-to-pixel, pixel-to-object, object-to-object), self-attention to complement CNNs, powerful operation because of adaptive computation, unified modeling between vision and language, and scalability in both model and data. The general computation of the self-attention operation is shown in Fig. 12(a).

In this section, we will review visual Transformer-based methods in the last two years. In literature, Transformer-based object detection is grouped into two kinds: Transformer-based detection head and Transformer-based feature extraction. Fig. 10 summarily describes the key milestones in the progresses of the recent trends in solving the object detection problem.

### 5.1. Transformer-based detection head

In existing object detectors, the detection head includes two branches corresponding to classification and regression tasks, which maps high extracted feature dimension from backbone network to lower feature dimension for a specific task, i.e., generate classification scores and regressed offsets for hand-crafted anchor boxes or points. Conventional methods treat two tasks independently without leveraging the interaction between bounding box predictions, or global image features *vs.* objects. Table 5 shows the comparison of Transformer-based detection head in two aspects: key improvements and performance. DETR [156] is the first end-to-end methods that performs interaction learning through Transformer operation to reason about detection results without any specific hand-crafted assumptions. In this year, many methods improve DETR architecture in various aspects such as efficient self-attention design [188–191], object query improvement [192–194,157], Transformer encoder, decoder improvement [195–197], and unsupervised learning [198]. In the following content, we review some representative methods of original DETR and its improvements.

DETR (Detection Transformer) [156] introduces a new end-to-end object detection method that applies Transformer architecture to the detection head, which achieves promising performance with two-stage detector Faster R-CNN [74]. The overall network of DETR is shown and described in Fig. 11. The Transformer encoder is illustrated in Fig. 11 and the Transformer decoder network is similar to the Transformer encoder. In these networks, positional encoding is a matrix that has the same size as the input sequence, containing the relationship between tokens in the image/feature sequence about relative or absolute position of them. The reason for adding positional encoding with input tokens is that self-attention layers in Transformer is permutation-invariant [187] (lack of inductive bias due to no recurrence and convolution in vision Transformer). To encode the information of the order of image sequence, the positional encoding is supplemented with the input tokens. In the literature, there are two kinds of positional encoding: learnable positional encoding and fixed positional encoding based on sine and cosine functions. Generally, DETR presents a simple network architecture that combines the conventional CNNs and Transformer, and views object detection results as a direct set prediction based on bipartite matching between ground truth and prediction set. This detector assumes a set of object queries is responsible for determining object locations and is learnable during updating network parameters. Each object query interacts with the global image feature, aggregates the important features from other queries, and also adopts the relations of input and output of decoder through the attention mechanism. Therefore, DETR extracts adequate information of bounding boxes, which eliminates hand-crafted designs such as anchor box generation and NMS procedure.

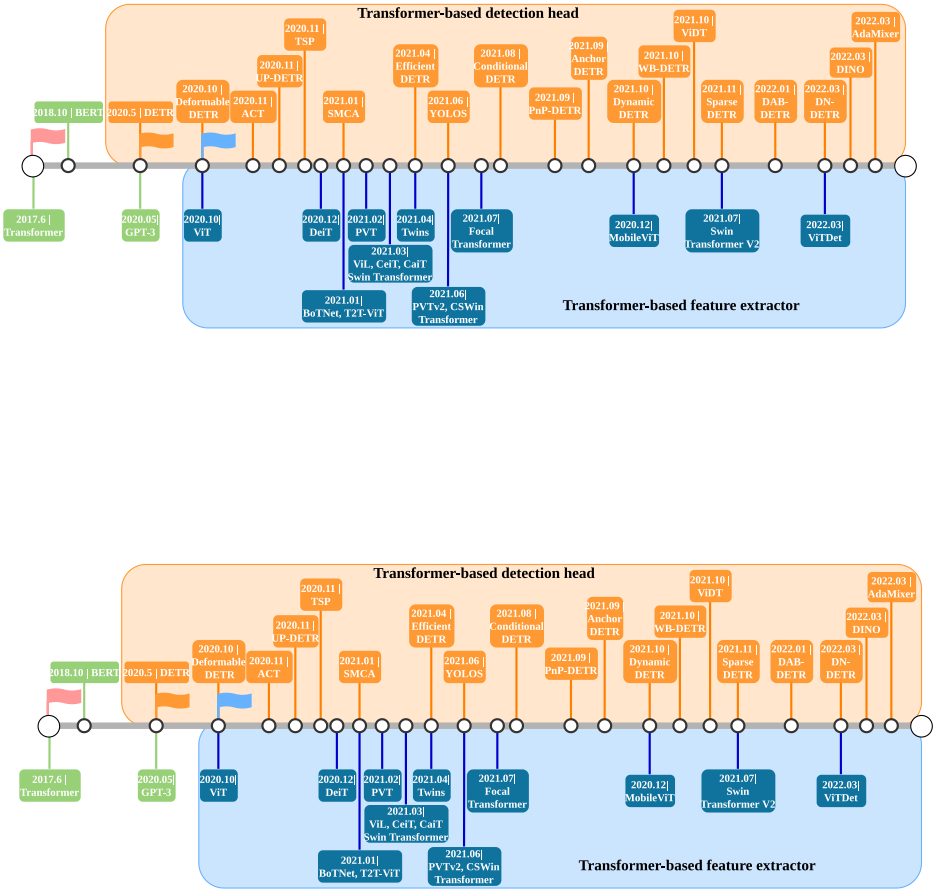



**Fig. 10.** The main milestones in the progress of the Transformer-based detection head and Transformer-based feature extraction.

**Table 5**
Comparative performance of DETR and its improvements on MS-COCO validation.

| Method | Baseline | Key improvements | Backbone | Schedule | *AP* | $AP^{50}$ | $AP^{75}$ | #params | GFLOPs | FPS |
|---|---|---|---|---|---|---|---|---|---|---|
| DETR [156] | – | – | ResNet-50 | 500 epochs | 42.0 | 62.4 | 44.2 | 41 | 86 | 28 |
| Deformable DETR [188] | DETR | - Multi-scale deformable self-attention<br>- Box refinement<br>- Two-stage network | ResNet-50 | 50 epochs | 46.2 | 65.2 | 50.0 | 40 | 173 | 19 |
| ACT-MTKD [189] | DETR | Adaptive clustering Transformer | ResNet-50 | fine-tuning (7 epochs) | 43.1 | – | – | – | 169 | – |
| SMCA [192] | DETR | Spatially modulated co-attention | ResNet-50 | 50 epochs | 43.7 | 63.6 | 47.2 | 40 | 152 | 10 |
| Conditional DETR [193] | DETR | Conditional cross-attention | ResNet-50 | 50 epochs | 40.9 | 61.8 | 43.3 | 44 | 90 | – |
| PnP-DETR [190] | DETR | Poll and pool sampling module | ResNet-50 | – | 41.8 | 62.1 | 44.4 | – | – | – |
| Dynamic DETR [191] | DETR | - Dynamic self-attention<br>- Dynamic cross-attention | ResNet-50 | 12 epochs | 42.9 | 61.0 | 46.3 | – | – | – |
| Efficient DETR [194] | DETR | Dense prior initialization | ResNet-50 | 36 epochs | 44.2 | 62.2 | 48.0 | 32 | 159 | – |
| DN-DETR [199] | DETR | Denoising training | ResNet-50 | 12 epochs | 41.7 | 61.4 | 44.1 | 44 | 216 | – |
| DINO [200] | DETR | - Contrastive denoising training<br>- Mixed query selection<br>- Box refinement | ResNet-50 | 12 epochs | 47.9 | 65.3 | 52.1 | 47 | 279 | 24 |
| AdaMixer [201] | DETR | - 3D feature sapce<br>- Adaptive mixing | ResNet-50 | 12 epochs | 44.1 | 63.1 | 47.8 | – | 132 | 24 |

Although DETR brings a simple and intuitive network architecture in detection literature, it has two drawbacks:

1. *Convergence speed*: DETR takes much longer training time than existing detectors (e.g., needs 500 epochs to get comparable accuracy) because self-attention and cross-attention blocks model the relations on large global context image from an initial dense set to a final sparse set, e.g., four parameters for each bounding box.
2. *Low detection performance on small objects*: DETR uses one-level feature with the lowest resolution of the backbone network for performing detection which relatively detects large objects on this feature. Previous detectors apply multi-level feature maps with various scales, where small objects are identified from feature maps with larger scales. However, DETR suffers too high computational cost when using multi-scale feature maps because the model complexity of the self-attention block increases quadratically with the input feature map resolution.

To overcome these problems, there are many methods to improve the self-attention module in the Transformer encoder-decoder and feature pyramid structure of DETR. Deformable DETR [188] proposes the deformable attention operation to define a learnable sparse sampling point set for key elements rather than using all feature map pixels. Self-attention and cross-attention modules in both the Transformer encoder and decoder are replaced with deformable attention. Cao et al. [192] replace the cross attention module in the Transformer decoder of DETR with Spatially Modulated Co-Attention (SMCA). SMCA performs element-wise multiplication between the learnable co-attention maps and object query weight maps. The co-attention maps model the long-range dependencies between object query and global image context. Object query weight map is generated by modeling the object query distribution as 2D spatial Gaussian. Meng et al. [193] propose Conditional DETR that learns a conditional spatial embedding for each object query. Based on the conditional spatial query, the cross attention in the Transformer decoder attends to object extreme points for bounding box regression and valid regions inside objects for leveraging the classification task. PnP-DETR [190] reduces computational costs in Transformer by eliminating redundant information of the global image context through a poll and pool (PnP) sampling module. To solve two drawbacks of DETR, Dai et al. [191] propose dynamic attention that applies to both the Transformer encoder and decoder, named Dynamic DETR. In the dynamic encoder, Dynamic DETR uses convolution operations to design different attention blocks estimating the self-attention mechanism. In the dynamic decoder, Dynamic DETR explores RoI-based dynamic attention which focuses on RoI features in a coarse-to-fine manner.

DN-DETR [199] discovers that the slow convergence speed of DETR originates from the unstableness of bipartite matching. To overcome this problem, DN-DETR adds noises to ground truth boxes and considers this signal as noised queries. These noised queries along with learnable anchor queries are attached to the Transformer decoder. According to DN-DETR, DINO [200] proposes three improvements: (1) a contrastive denoising training for both positive and negative samples mitigates ambiguous learning of the detectors due to occlusion problems, (2) a mixed query selection improves anchor query initialization based on positional encoding and top-k features, and (3) a box refinement is proposed to update box coordinates twice times.

AdaMixer [201] addresses the slow convergence of the detector DETR in two aspects: (1) multi-level features can be viewed as 3D

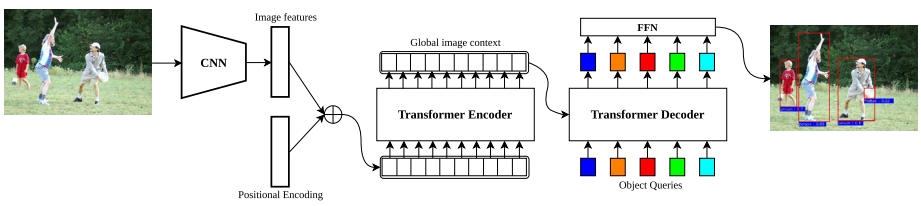



**Fig. 11.** Firstly, DETR utilizes the CNN network to perform feature extraction from the input image and produces a low spatial resolution feature map. Secondly, this feature map is flattened to the image feature vector, which is suitable for the Transformer computation (e.g., this vector is considered as a sequence of tokens). The positional encoding is added with the feature vector to serve as input of the Transformer. Thirdly, the Transformer encoder model the relationship between a token and other tokens, and outputs the global image context. Fourthly, the Transformer decoder reasons about the relations of learnable object queries and the global contextual feature. Fifthly, FFNs are feed-forward networks prediction, designed as classification and regression branches to directly generate the final prediction set of class scores and bounding box coordinates.

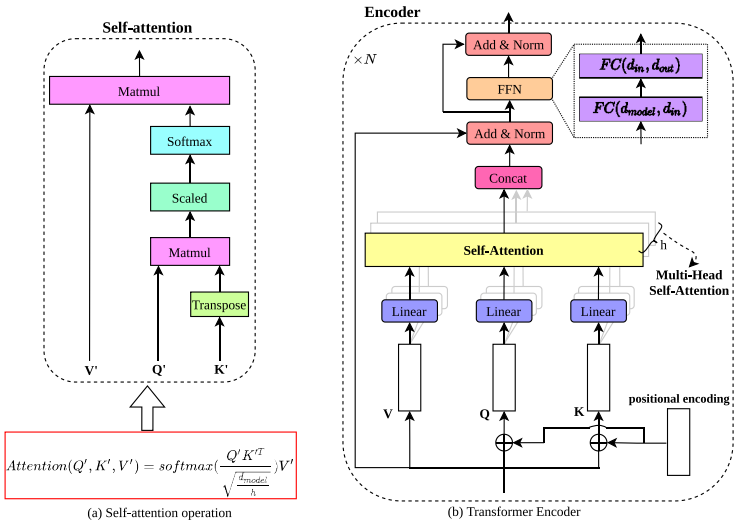



**Fig. 12.** (a) Self-attention operation is addressed, where $Q, K, V$ are query matrix, key, and value matrix, respectively. $d_{model}$ is the input embedding dimension, and $h$ indicates a number of self-attention blocks in one multi-head self-attention module. (b) Transformer encoder is illustrated, where Linear is Fully-connected layer to map matrix $Q, K, V$ to matrix $Q\prime, K\prime, V\prime$. FFN is the feed-forward network including two consecutive fully-connect layers (FC layers).

feature space and based on this design, the Adamixer decoder can select features that rely on object's variation, (2) an adaptive mixing uses spatial and channel mixing in MLP-mixer [202] to decode object queries.

### 5.2. Transformer-based feature extraction

In recent years, many vision Transformer-based backbones have been proposed for the image classification task such as [203–209], which achieve promising improvements. However, object detector requires high-dimension features and multi-scale features to perform dense predictions while vision Transformer has high complexity on large dimensions and only outputs a single low-resolution feature. To leverage Transformer's strong global modeling capability, many studies try to incorporate Transformer as feature extraction into object detection network by constructing pyramid vision Transformer [210,211], and efficient self-attention computation [212–215]. Instead of applying the Transformer architecture to the detection head, state-of-the-art object detectors try to utilize the Transformer as feature extraction in the backbone network. Fig. 13 illustrates the general architecture of Transformer-based feature extraction where all stage is the same architecture and Fig. 14 describes four types of Transformer Encoder. Originally, the Transformer [187] used in Natural Language Processing (NLP) tasks requires a 1D sequence of tokens. To process the high dimension of visual data, patch embedding is proposed to

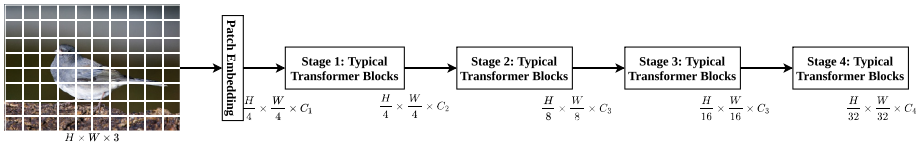



**Fig. 13.** The general architecture of Transformer-based feature extraction. Firstly, Patch Embedding splits the input image into patches (a sequence of tokens). Secondly, Typical Transformer Blocks are SRA [210], Swin Transformer Block [212], and Focal Transformer Block [213], which model the relations of patches and produce the global contextual features.

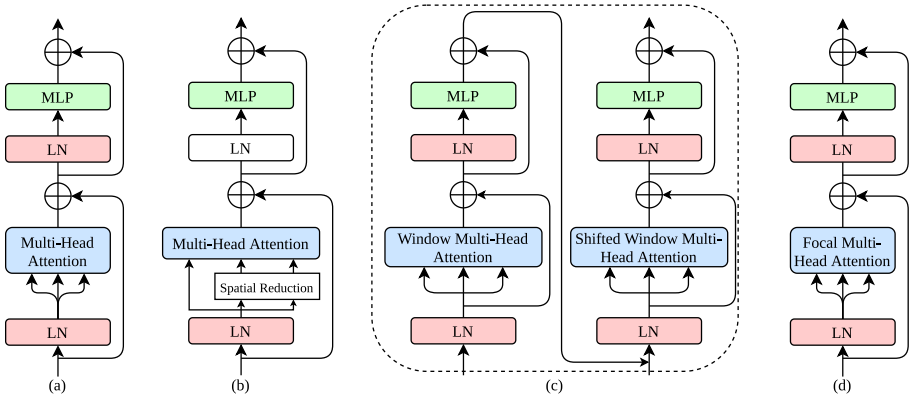



**Fig. 14.** Four types of Transformer Encoder: (a) Stand Transformer Encoder in ViT [203], (b) Spatial-Reduction Attention in PVT [210], (c) Swin Transformer Block [212], and (d) Focal Transformer [213].

separate input images/features into a sequence of flattened patches. Given the image feature $F_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ at stage $i$, the output of the patch embedding is the patch feature $F_i^p \in \mathbb{R}^{N \times (P^2 C_i)}$, where $H_i, W_i, C_i$ are the height, width, and channels of the feature map, $N = (H_i W_i)/P^2$ is the number of patches, and $P$ is the patch size. In the common implementations [203,210,212], patch embedding is performed by a convolution layer with kernel size $P$ and stride $P$.

PVT [210] empowers Vision Transformer (ViT) [203] originally designed for the image classification task to dense object detection by employing the feature pyramid from the conventional CNNs. The combination between ViT and feature pyramid network is called Pyramid Vision Transformer (PVT) that served as a versatile backbone for many dense predictions. The difference between ViT and PVT is the feature map resolution. The size of ViT's output is similar to the input size and thus, all stages in the network use single-scale. It leads to high model complexity when applying the ViT-based backbone to dense predictions that require high resolution and multi-scale feature maps. PVT solves these problems by constructing the feature pyramid with different scales, and this structure can generate global receptive fields suitable for the object detection task. Fig. 14(a) shows the Spatial-Reduction Attention (SRA) in PVT that reduces the spatial dimension of Key Matrix and Value matrix via spatial reduction operation using convolution layer with different strides.

Liu et al. [212] propose the Swin Transformer that is a simple and efficient operation. Swin Transformer uses shifted window partition between neighborhood self-attention modules instead of the sliding window in ViT and PVT. Fig. 14(c) shows the swin transformer block including two important components: Window Multi-Head Attention and Shifted Window Multi-Head Attention. Both attentions are calculated within local windows based on a hierarchical manner. And a shifted window partitioning method models the relations between non-overlapping windows. Moreover, the number of patches in each window is not changed. Therefore, the model complexity of the Swin Transformer is linear with input size.

Yang et al. [213] introduce the Focal Transformer in which Focal self-attention with less computational costs is a key element that models fine-grained local and coarse-grained global dependencies in visual data. Firstly, the Focal Transformer partitions the input feature map into multiple windows on multi-scale feature maps, and tokens are share the same set of neighborhood regions. Secondly, multiple levels of tokens are concatenated to calculate the Key and Value Matrices. Finally, the Focal Transformer computes attentions based on standard Multi-head self-attention.

Nowadays, Transformer-based feature extraction and detection head have dominated in solving object detection. Soft Teacher [216] uses Swin Transformer-based backbone, which achieves the best detection performance on MS-COCO dataset (e.g., 61.3% AP) versus existing methods. Many researchers have applied DETR and Deformable DETR for solving other tasks such as multiple object tracking [217–220], instance segmentation [221–223].

## 6. Open issues

**Anchor Assignment Methods.** Many methods to separate prior anchor boxes into positive samples and negative samples for efficiently training detectors have been introduced, analyzed in two groups: hard anchor assignment and soft anchor assignment. However, recent researches leave some unsolved issues that require more investigation:

1. Most of the advanced anchor assignments [153,161,154,164] relies on prior assumption such as prior center, IoU-based method, or $L1$ distance to construct a positive bag. This strategy points out some phenomena: (i) Defining hard thresholds requires more experiments and time-consuming; (ii) It produces a large number of noisy anchor boxes/anchor points for the next step.
2. Common methods apply the linear combination of classification and localization losses for constructing the cost matrix. However, it leads to objective imbalance due to the different properties of multi-task learning such as: (i) The range and gradient norm of different losses might be different; (ii) The difficulty of each task is different.
3. Common anchor assignment methods weakly take challenging problems such as occlusion, ambiguity, and irregular objects into account. It leads to ambiguous learning during optimizing the model and degraded performance.

**Sampling Methods.** In recent years, many methods have been proposed to determine the usefulness of the positive samples and negative samples according to many criteria such as hard example mining, IoU score, and confidence score. Although these methods achieve significant improvements, there are some open issues that need more studies:

1. Defining the criterion of sample weighting needs more consideration. Some methods use classification scores to compute sample weighting, unexpectedly outputting many outliers during sampling and hindering the detection model. Unifying all criteria (classification score, IoU score, classification loss, and localization loss) into one sampling metric to generate optimal sample weighting needs more investigation.
2. There is an inconsistency between focusing on easy samples [94,143] and focusing on hard samples [97,91] during training.
3. Most of the soft sampling methods only control the contribution of useful samples to the classification loss. However, object detection solves classification and localization tasks simultaneously, and only considering the classification term during sampling leads to objective imbalance. Therefore, reshaping the localization loss based on sample weighting opens challenging problems.

**Transformer-based object detection.** Transformer-based object detection has become a new trend in the object detection community, achieving fast progresses during recent years and outperforming CNN-based object detection in both performance and efficiency. According to the analysis in Section V, Transformer-based object detection needs further studies:

1. The combination of vision Transformer and CNN's architecture lacks investigation. Since self-attention operation in Transformer complements to CNNs, only several methods try to integrate Transformer into conventional CNNs networks.
2. Designing Transformer architecture based on CNNs' characteristics such as hierarchical property [120], dense connection [176], and high-resolution network [179] can improve performance.
3. The design of random object queries is arbitrary and unsuitable for object detection, i.e., difficult to optimize. Object queries are computed based on prior anchor points or anchor boxes that need more studies by researchers.
4. One-to-one matching in DETR and its variants can not solve ambiguous anchors that are aforementioned in Section IV.
5. Designing light-weight Transformer architectures [224] for mobile devices or embedded devices is still open.

## 7. Conclusion

This paper presents a thorough review of the main components: anchor assignment and sample sampling in object detection net-

works. State-of-the-art detectors heavily rely on these components to efficiently train the detection network. In order to provide a larger view, we grouped the advanced methods in a problem-based taxonomy and its solutions. Based on the problem-based taxonomy, each method is discussed and analyzed systematically. Besides, we identified the advantages and disadvantages of each problem in-depth, and introduced research issues. Moreover, we provided the recent trends in object detection that modern detectors entirely apply vision Transformer operation to detection network architecture in which Transformer-based feature extractor and Transformer-based detection head have been attracted much attention from object detection researchers during the last two years. We hope the object detection community can identify the current status of object detection methods to propose better solutions to anchor assignment, sample sampling, and Transformer-based methods in object detection research.

## 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.

## Acknowledgment

This work was supported by "Region Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE)(2021RIS-003).

## References

[1] X.-T. Vo, T.-D. Tran, D.-L. Nguyen, K.-H. Jo, Regression-aware classification feature for pedestrian detection and tracking in video surveillance systems, in: International Conference on Intelligent Computing, Springer, 2021, pp. 816–828.
[2] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Fu, Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking, in: European Conference on Computer Vision, Springer, 2020, pp. 145–161.
[3] Z. Wang, L. Zheng, Y. Liu, Y. Li, S. Wang, Towards real-time multi-object tracking, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer, 2020, pp. 107–122.
[4] X. Zhou, V. Koltun, P. Krähenbühl, Tracking objects as points, European Conference on Computer Vision, Springer (2020) 474–490.
[5] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, Fairmot: On the fairness of detection and re-identification in multiple object tracking, Int. J. Comput. Vision (2021) 1–19.
[6] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu, X. Wang, Bytetrack: Multi-object tracking by associating every detection box, arXiv preprint arXiv:2110.06864.
[7] H. Duan, Y. Zhao, K. Chen, D. Shao, D. Lin, B. Dai, Revisiting skeleton-based action recognition, arXiv preprint arXiv:2104.13586.
[8] Y. Obinata, T. Yamamoto, Temporal extension module for skeleton-based action recognition, in: 2020 25th International Conference on Pattern Recognition (ICPR) IEEE, 2021, pp. 534–540.
[9] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L.S. Davis, J. Kautz, Step: Spatio-temporal progressive learning for video action detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 264–272.
[10] Y. Li, Z. Wang, L. Wang, G. Wu, Actions as moving points, European Conference on Computer Vision, Springer (2020) 68–84.
[11] Y. Yan, J. Li, J. Qin, S. Liao, X. Yang, Efficient person search: An anchor-free approach, arXiv preprint arXiv:2109.00211.
[12] Y. Yan, J. Li, J. Qin, S. Bai, S. Liao, L. Liu, F. Zhu, L. Shao, Anchor-free person search, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7690–7699.
[13] Z. Li, D. Miao, Sequential end-to-end network for efficient person search, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 2011–2019.
[14] V.-T. Hoang, D.-S. Huang, K.-H. Jo, 3-d facial landmarks detection for intelligent video systems, IEEE Trans. Industr. Inf. 17 (1) (2020) 578–586.
[15] V.-T. Hoang, K.-H. Jo, 3-d human pose estimation using cascade of multiple neural networks, IEEE Trans. Industr. Inf. 15 (4) (2018) 2064–2072.
[16] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.
[17] T.-D. Tran, X.-T. Vo, M.-A. Russo, K.-H. Jo, Simple fine-tuning attention modules for human pose estimation, International Conference on Computational Collective Intelligence, Springer (2020) 175–185.
[18] X.-T. Vo, T.-D. Tran, D.-L. Nguyen, K.-H. Jo, Dynamic multi-loss weighting for multiple people tracking in video surveillance systems, in: 2021 IEEE 19th International Conference on Industrial Informatics (INDIN) IEEE, 2021, pp. 1–6.
[19] Z. Fu, Y. Chen, H. Yong, R. Jiang, L. Zhang, X.-S. Hua, Foreground gating and background refining network for surveillance object detection, IEEE Trans. Image Process. 28 (12) (2019) 6077–6090.
[20] A. Shahbaz, K.-H. Jo, Deep atrous spatial features-based supervised foreground detection algorithm for industrial surveillance systems, IEEE Trans. Industr. Inf. 17 (7) (2020) 4818–4826.
[21] Q. Fan, L. Brown, J. Smith, A closer look at faster r-cnn for vehicle detection, in: 2016 IEEE intelligent vehicles symposium (IV), IEEE, 2016, pp. 124–129
[22] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, IEEE conference on computer vision and pattern recognition, IEEE 2012 (2012) 3354–3361.
[23] X. Dai, Hybridnet: A fast vehicle detection system for autonomous driving, Signal Process.: Image Commun. 70 (2019) 79–88.
[24] C.-T. Lin, P. Sherryl Santoso, S.-P. Chen, H.-J. Lin, S.-H. Lai, Fast vehicle detector for autonomous driving, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 222–229.
[25] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: A review, IEEE Trans. Pattern Anal. Mach. Intell. 28 (5) (2006) 694–711.
[26] Y. Aoki, H. Goforth, R.A. Srivatsan, S. Lucey, Pointnetlk: Robust & efficient point cloud registration using pointnet, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7163–7172.
[27] G. Du, K. Wang, S. Lian, K. Zhao, Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: a review, Artif. Intell. Rev. 54 (3) (2021) 1677–1734.
[28] U. Asif, J. Tang, S. Harrer, Graspnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices., in: IJCAI, Vol. 7, 2018, pp. 4875–4882.
[29] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, et al., Using simulation and domain adaptation to improve efficiency of deep robotic grasping, IEEE international conference on robotics and automation (ICRA), IEEE 2018 (2018) 4243–4250.
[30] D. Chen, J. Li, Z. Wang, K. Xu, Learning canonical shape space for category-level 6d object pose and size estimation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11973–11982.
[31] B. Li, Z.-T. Fan, X.-L. Zhang, D.-S. Huang, Robust dimensionality reduction via feature space to feature space distance metric learning, Neural Networks 112 (2019) 1–14.
[32] C.-Y. Lu, D.-S. Huang, Optimized projections for sparse representation based classification, Neurocomputing 113 (2013) 213–219.
[33] R.-X. Hu, W. Jia, D.-S. Huang, Y.-K. Lei, Maximum margin criterion with tensor representation, Neurocomputing 73 (10–12) (2010) 1541–1549.
[34] F. Han, D.-S. Huang, Z.-H. Zhu, T.-H. Rong, The forecast of the postoperative survival time of patients suffered from non-small cell lung cancer based on pca and extreme learning machine, Int. J. Neural Syst. 16 (01) (2006) 39–46.
[35] Q.-H. Ling, Y.-Q. Song, F. Han, D. Yang, D.-S. Huang, An improved ensemble of random vector functional link networks based on particle swarm optimization with double optimization strategy, Plos One 11 (11) (2016) e0165803.
[36] L. Zhu, D.-S. Huang, A rayleigh–ritz style method for large-scale discriminant analysis, Pattern Recogn. 47 (4) (2014) 1698–1708.
[37] L. Zhu, D.-S. Huang, Efficient optimally regularized discriminant analysis, Neurocomputing 117 (2013) 12–21.
[38] B. Li, C. Wang, D.-S. Huang, Supervised feature extraction based on orthogonal discriminant projection, Neurocomputing 73 (1–3) (2009) 191–196.
[39] B. Li, D.-S. Huang, C. Wang, K.-H. Liu, Feature extraction using constrained maximum variance mapping, Pattern Recogn. 41 (11) (2008) 3287–3294.
[40] F. Han, Q.-H. Ling, D.-S. Huang, An improved approximation approach incorporating particle swarm optimization and a priori information into neural networks, Neural Comput. Appl. 19 (2) (2010) 255–261.
[41] D.-S. Huang, J.-X. Du, A constructive hybrid structure optimization methodology for radial basis probabilistic neural networks, IEEE Trans. Neural Networks 19 (12) (2008) 2099–2115.
[42] J.-X. Du, D.-S. Huang, G.-J. Zhang, Z.-F. Wang, A novel full structure optimization algorithm for radial basis probabilistic neural networks, Neurocomputing 70 (1–3) (2006) 592–596.
[43] J. Zhang, D.-S. Huang, T.-M. Lok, M.R. Lyu, A novel adaptive sequential niche technique for multimodal function optimization, Neurocomputing 69 (16–18) (2006) 2396–2401.
[44] F. Han, D.-S. Huang, A new constrained learning algorithm for function approximation by encoding a priori information into feedforward neural networks, Neural Comput. Appl. 17 (5) (2008) 433–439.
[45] F. Han, Q.-H. Ling, D.-S. Huang, Modified constrained learning algorithms incorporating additional functional constraints into neural networks, Inf. Sci. 178 (3) (2008) 907–919.

[46] F. Han, D.-S. Huang, Improved constrained learning algorithms by incorporating additional functional constraints into neural networks, Appl. Math. Comput. 174 (1) (2006) 34–50.
[47] Z.-Q. Zhao, D.-S. Huang, A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability, Appl. Math. Model. 31 (7) (2007) 1271–1281.
[48] J.-X. Du, D.-S. Huang, X.-F. Wang, X. Gu, Shape recognition based on neural networks trained by differential evolution algorithm, Neurocomputing 70 (4–6) (2007) 896–903.
[49] C.-H. Zheng, D.-S. Huang, Z.-L. Sun, M.R. Lyu, T.-M. Lok, Nonnegative independent component analysis based on minimizing mutual information technique, Neurocomputing 69 (7–9) (2006) 878–883.
[50] W.-B. Zhao, D.-S. Huang, J.-Y. Du, L.-M. Wang, Genetic optimization of radial basis probabilistic neural networks, Int. J. Pattern Recognit Artif Intell. 18 (08) (2004) 1473–1499.
[51] D.-S. Huang, S.-D. Ma, Linear and nonlinear feedforward neural network classifiers: a comprehensive understanding, J. Intell. Syst. 9 (1) (1999) 1–38.
[52] D.-S. Huang, Radial basis probabilistic neural networks: Model and application, Int. J. Pattern Recognit Artif Intell. 13 (07) (1999) 1083–1101.
[53] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), Vol. 1, Ieee, 2005, pp. 886–893
[54] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: 2008 IEEE conference on computer vision and pattern recognition, Ieee, 2008, pp. 1–8
[55] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the seventh IEEE international conference on computer vision, Vol. 2, Ieee, 1999, pp. 1150–1157.
[56] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.
[57] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Trans. Pattern Anal. Mach. Intell. 24 (4) (2002) 509–522.
[58] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. 25 (2012) 1097–1105.
[59] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, Deep learning for generic object detection: A survey, Int. J. Comput. Vision 128 (2) (2020) 261–318.
[60] Z. Zou, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, arXiv preprint arXiv:1905.05055.
[61] S. Agarwal, J.O.D. Terrail, F. Jurie, Recent advances in object detection in the age of deep convolutional neural networks, arXiv preprint arXiv:1809.03193.
[62] X. Wu, D. Sahoo, S.C. Hoi, Recent advances in deep learning for object detection, Neurocomputing 396 (2020) 39–64.
[63] Z.-Q. Zhao, P. Zheng, S.-T. Xu, X. Wu, Object detection with deep learning: a review, IEEE Trans. Neural Networks Learn. Syst. 30 (11) (2019) 3212–3232.
[64] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
[65] K. Oksuz, B.C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: A review, IEEE transactions on pattern analysis and machine intelligence.
[66] Y. Zhou, L. Liu, L. Shao, M. Mellor, Dave: A unified framework for fast vehicle detection and annotation, in: European conference on computer vision, Springer, 2016, pp. 278–293.
[67] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: A review, IEEE Trans. Pattern Anal. Mach. Intell. 28 (5) (2006) 694–711.
[68] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Comput. Vis. Image Underst. 138 (2015) 1–24.
[69] A. Kumar, A. Kaur, M. Kumar, Face detection techniques: a review, Artif. Intell. Rev. 52 (2) (2019) 927–948.
[70] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.
[71] J. Cao, Y. Pang, J. Xie, F.S. Khan, L. Shao, From handcrafted to deep features for pedestrian detection: a survey, IEEE transactions on pattern analysis and machine intelligence.
[72] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4073–4082.
[73] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[74] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Adv. Neural Inform. Process. Syst. 28 (2015) 91–99.
[75] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vision 104 (2) (2013) 154–171.
[76] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Advances in neural information processing systems, 2016, pp. 379–387.
[77] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: European conference on computer vision, Springer, 2016, pp. 354–370.
[78] H. Lee, S. Eum, H. Kwon, Me r-cnn: multi-expert region-based cnn for object detection, in: ICCV, 2017.
[79] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
[80] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6054–6063.
[81] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, Y. Fu, Rethinking classification and localization for object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10186–10195.
[82] S. Qiao, L.-C. Chen, A. Yuille, Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10213–10224.
[83] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, L. Zhang, Dynamic head: Unifying object detection heads with attentions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7373–7382.
[84] S. Bell, C.L. Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2874–2883.
[85] 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 IEEE conference on computer vision and pattern recognition, 2018, pp. 6985–6994.
[86] Z. Chen, S. Huang, D. Tao, Context refinement for object detection, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 71–86.
[87] A. Shrivastava, A. Gupta, Contextual priming and feedback for faster r-cnn, in: European conference on computer vision, Springer, 2016, pp. 330–348.
[88] Z. Qin, Z. Li, Z. Zhang, Y. Bao, G. Yu, Y. Peng, J. Sun, Thundernet: Towards real-time generic object detection on mobile devices, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6718–6727.
[89] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, Gcnet: Non-local networks meet squeeze-excitation networks and beyond, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.
[90] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.
[91] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 761–769.
[92] X. Wang, A. Shrivastava, A. Gupta, A-fast-rcnn: Hard positive generation via adversary for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2606–2615.
[93] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2888–2897.
[94] Y. Cao, K. Chen, C.C. Loy, D. Lin, Prime sample attention in object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11583–11591.
[95] 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: European conference on computer vision, Springer, 2014, pp. 740–755.
[96] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.
[97] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.
[98] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: Deconvolutional single shot detector, arXiv preprint arXiv:1701.06659.
[99] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 385–400.
[100] T. Wang, R.M. Anwer, H. Cholakkal, F.S. Khan, Y. Pang, L. Shao, Learning rich features at high-speed for single-shot object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1971–1980.
[101] R.J. Wang, X. Li, C.X. Ling, Pelee: A real-time object detection system on mobile devices, arXiv preprint arXiv:1804.06882.
[102] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
[103] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7263–7271.
[104] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767.
[105] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934.
[106] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, Scaled-yolov4: Scaling cross stage partial network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13029–13038.

[107] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[108] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.
[109] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 734–750.
[110] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, arXiv preprint arXiv:1904.07850.
[111] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6569–6578.
[112] X. Zhou, J. Zhuo, P. Krahenbuhl, Bottom-up object detection by grouping extreme and center points, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 850–859.
[113] A. Newell, Z. Huang, J. Deng, Associative embedding: End-to-end learning for joint detection and grouping, arXiv preprint arXiv:1611.05424.
[114] Z. Yang, S. Liu, H. Hu, L. Wang, S. Lin, Reppoints: Point set representation for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9657–9666.
[115] J. Wang, K. Chen, S. Yang, C.C. Loy, D. Lin, Region proposal by guided anchoring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2965–2974.
[116] C. Zhu, Y. He, M. Savvides, Feature selective anchor-free module for single-shot object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 840–849.
[117] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, J. Shi, Foveabox: Beyound anchor-based object detection, IEEE Trans. Image Process. 29 (2020) 7389–7398.
[118] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9627–9636.
[119] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.
[120] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[121] 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, arXiv preprint arXiv:1704.04861.
[122] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8759–8768.
[123] G. Ghiasi, T.-Y. Lin, Q.V. Le, Nas-fpn, Learning scalable feature pyramid architecture for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7036–7045.
[124] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, J. Sun, You only look one-level feature, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13039–13048.
[125] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[126] X.-T. Vo, K.-H. Jo, Enhanced feature pyramid networks by feature aggregation module and refinement module, in: 2020 13th International Conference on Human System Interaction (HSI) IEEE, 2020, pp. 63–67.
[127] X.-T. Vo, L. Wen, T.-D. Tran, K.-H. Jo, Bidirectional non-local networks for object detection, International Conference on Computational Collective Intelligence, Springer (2020) 491–501.
[128] H. Zhang, H. Chang, B. Ma, N. Wang, X. Chen, Dynamic r-cnn: Towards high quality object detection via dynamic training, European Conference on Computer Vision, Springer (2020) 260–275.
[129] T. Vu, H. Jang, T.X. Pham, C.D. Yoo, Cascade rpn: Delving into high-quality region proposal network with adaptive convolution, in: Conference on Neural Information Processing Systems (NeurIPS), 2019.
[130] X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid r-cnn, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7363–7372.
[131] X. Lu, B. Li, Y. Yue, Q. Li, J. Yan, Grid r-cnn plus: Faster and better, arXiv preprint arXiv:1906.05688.
[132] Z. Huang, L. Huang, Y. Gong, C. Huang, X. Wang, Mask scoring r-cnn, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6409–6418.
[133] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, P. Dollár, Designing network design spaces, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10428–10436.
[134] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, P.H. Torr, Res2net: A new multi-scale backbone architecture, IEEE transactions on pattern analysis and machine intelligence.
[135] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, et al., Resnest: Split-attention networks, arXiv preprint arXiv:2004.08955.
[136] X.-T. Vo, T.-D. Tran, D.-L. Nguyen, K.-H. Jo, Stair-step feature pyramid networks for object detection, in: International Workshop on Frontiers of Computer Vision, Springer, 2021, pp. 168–175.
[137] J. Wang, W. Zhang, Y. Cao, K. Chen, J. Pang, T. Gong, J. Shi, C.C. Loy, D. Lin, Side-aware boundary localization for more precise object detection, in: European Conference on Computer Vision, Springer, 2020, pp. 403–419.
[138] K. Chen, W. Lin, J. Li, J. See, J. Wang, J. Zou, Ap-loss for accurate one-stage object detection, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3782–3798.
[139] Q. Qian, L. Chen, H. Li, R. Jin, Dr loss: Improving object detection by distributional ranking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12164–12172.
[140] S. Zhang, C. Chi, Y. Yao, Z. Lei, S.Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9759–9768.
[141] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection, in: NeurIPS, 2020.
[142] X. Li, W. Wang, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11632–11641.
[143] H. Zhang, Y. Wang, F. Dayoub, N. Sunderhauf, Varifocalnet: An iou-aware dense object detector, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8514–8523.
[144] K. Oksuz, B.C. Cam, E. Akbas, S. Kalkan, A ranking-based, balanced loss function unifying classification and localisation in object detection, in: Advances in Neural Information Processing Systems (NeurIPS), 2020.
[145] K. Oksuz, B.C. Cam, E. Akbas, S. Kalkan, Rank & sort loss for object detection and instance segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3009–3018.
[146] Z. Chen, C. Yang, Q. Li, F. Zhao, Z.-J. Zha, F. Wu, Disentangle your dense object detector, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 4939–4948.
[147] X.-T. Vo, K.-H. Jo, Accurate bounding box prediction for single-shot object detection, IEEE Transactions on Industrial Informatics.
[148] X. Zhang, F. Wan, C. Liu, R. Ji, Q. Ye, FreeAnchor: Learning to match anchors for visual object detection, Neural Inform. Process. Syst. (2019).
[149] H. Li, Z. Wu, C. Zhu, C. Xiong, R. Socher, L.S. Davis, Learning from noisy anchors for one-stage object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10588–10597.
[150] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu, D. Huang, Multiple anchor learning for visual object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10206–10215.
[151] K. Kim, H.S. Lee, Probabilistic anchor assignment with iou prediction for object detection, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16, Springer, 2020, pp. 355–371.
[152] B. Zhu, J. Wang, Z. Jiang, F. Zong, S. Liu, Z. Li, J. Sun, Autoassign: Differentiable label assignment for dense object detection, arXiv preprint arXiv:2007.03496.
[153] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, N. Zheng, End-to-end object detection with fully convolutional network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15849–15858.
[154] Z. Ge, S. Liu, Z. Li, O. Yoshie, J. Sun, Ota: Optimal transport assignment for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 303–312.
[155] C.H. Nguyen, T.C. Nguyen, T.N. Tang, N.L. Phan, Improving object detection by label assignment distillation, arXiv preprint arXiv:2108.10520.
[156] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, European Conference on Computer Vision, Springer (2020) 213–229.
[157] Y. Wang, X. Zhang, T. Yang, J. Sun, Anchor detr: Query design for transformer-based detector, arXiv preprint arXiv:2109.07107.
[158] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, L. Zhang, DAB-DETR: Dynamic anchor boxes are better queries for DETR, in: International Conference on Learning Representations, 2022.
[159] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.
[160] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al., Sparse r-cnn: End-to-end object detection with learnable proposals, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14454–14463.
[161] P. Sun, Y. Jiang, E. Xie, W. Shao, Z. Yuan, C. Wang, P. Luo, What makes for end-to-end object detection?, in: International Conference on Machine Learning PMLR, 2021, pp 9934–9944.
[162] Z. Ge, J. Wang, X. Huang, S. Liu, O. Yoshie, Lla: Loss-aware label assignment for dense pedestrian detection, Neurocomputing 462 (2021) 272–281.
[163] Z. Gao, L. Wang, G. Wu, Mutual supervision for dense object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3641–3650.
[164] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, Yolox: Exceeding yolo series in 2021, arXiv preprint arXiv:2107.08430.

[165] C. Feng, Y. Zhong, Y. Gao, M.R. Scott, W. Huang, Tood: Task-aligned one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3510–3519.
[166] J. Chen, B. Luo, Q. Wu, J. Chen, X. Peng, Overlap sampler for region-based object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 767–775.
[167] J. Chen, D. Liu, T. Xu, S. Wu, Y. Cheng, E. Chen, Is heuristic sampling necessary in training deep object detectors?, IEEE Trans Image Process. 30 (2021) 8454–8467.
[168] B. Li, Y. Liu, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8577–8584.
[169] Q. Cai, Y. Pan, Y. Wang, J. Liu, T. Yao, T. Mei, Learning a unified sample weighting network for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14173–14182.
[170] S. Li, C. He, R. Li, L. Zhang, A dual weighting label assignment scheme for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[171] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7482–7491.
[172] K. Oksuz, B.C. Cam, E. Akbas, S. Kalkan, Rank & sort loss for object detection and instance segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3009–3018.
[173] J. Tan, X. Lu, G. Zhang, C. Yin, Q. Li, Equalization loss v2: A new gradient balance approach for long-tailed object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1685–1694.
[174] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, J. Yan, Equalization loss for long-tailed object recognition, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11662–11671.
[175] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[176] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[177] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
[178] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, International Conference on Machine Learning, PMLR (2019) 6105–6114.
[179] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, J. Wang, Lite-hrnet: A lightweight high-resolution network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10440–10450.
[180] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, Repvgg: Making vgg-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.
[181] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
[182] S. Woo, J. Park, J.-Y. Lee, I.S. Kweon, Cbam: Convolutional block attention module, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
[183] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.
[184] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, Eca-net: Efficient channel attention for deep convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
[185] Z. Qin, P. Zhang, F. Wu, X. Li, Fcanet: Frequency channel attention networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 783–792.
[186] Q. Hou, D. Zhou, J. Feng, in: Coordinate attention for efficient mobile network design, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13713–13722.
[187] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[188] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object eetection, in: International Conference on Learning Representations, 2021.
[189] M. Zheng, P. Gao, X. Wang, H. Li, H. Dong, End-to-end object detection with adaptive clustering transformer, arXiv preprint arXiv:2011.09315.
[190] T. Wang, L. Yuan, Y. Chen, J. Feng, S. Yan, Pnp-detr: Towards efficient visual analysis with transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4661–4670.
[191] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, L. Zhang, Dynamic detr: End-to-end object detection with dynamic attention, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2988–2997.
[192] P. Gao, M. Zheng, X. Wang, J. Dai, H. Li, Fast convergence of detr with spatially modulated co-attention, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3621–3630.
[193] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, J. Wang, Conditional detr for fast training convergence, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3651–3660.
[194] Z. Yao, J. Ai, B. Li, C. Zhang, Efficient detr: Improving end-to-end object detector with dense prior, arXiv preprint arXiv:2104.01318.
[195] Z. Sun, S. Cao, Y. Yang, K.M. Kitani, Rethinking transformer-based set prediction for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3611–3620.
[196] Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, W. Liu, You only look at one sequence: Rethinking transformer in vision through object detection, arXiv preprint arXiv:2106.00666.
[197] F. Liu, H. Wei, W. Zhao, G. Li, J. Peng, Z. Li, Wb-detr: Transformer-based detector without backbone, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2979–2987.
[198] Z. Dai, B. Cai, Y. Lin, J. Chen, Up-detr: Unsupervised pre-training for object detection with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1601–1610.
[199] F. Li, H. Zhang, S. Liu, J. Guo, L.M. Ni, L. Zhang, Dn-detr: Accelerate detr training by introducing query denoising.
[200] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L.M. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection, arXiv preprint arXiv:2203.03605.
[201] Z. Gao, L. Wang, B. Han, S. Guo, Adamixer: A fast-converging query-based object detector.
[202] I.O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al., Mlp-mixer: An all-mlp architecture for vision, Advances in Neural Information Processing Systems 34.
[203] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021.
[204] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, A. Vaswani, Bottleneck transformers for visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16519–16529.
[205] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning PMLR, 2021, pp. 10347–10357.
[206] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, W. Wu, Incorporating convolution designs into visual transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 579–588.
[207] X. Chu, Z. Tian, Y. Wang, B. Zhang, H. Ren, X. Wei, H. Xia, C. Shen, Twins: Revisiting the design of spatial attention in vision transformers, arXiv preprint arXiv:2104.13840 1 (2) (2021) 3.
[208] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, H. Jégou, Going deeper with image transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 32–42.
[209] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F.E. Tay, J. Feng, S. Yan, Tokens-to-token vit: Training vision transformers from scratch on imagenet, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 558–567.
[210] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 568–578.
[211] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pvtv 2: Improved baselines with pyramid vision transformer, arXiv preprint arXiv:2106.13797.
[212] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 10012–10022.
[213] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, J. Gao, Focal self-attention for local-global interactions in vision transformers, arXiv preprint arXiv:2107.00641.
[214] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, J. Gao, Multi-scale vision longformer: A new vision transformer for high-resolution image encoding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 2998–3008.
[215] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, B. Guo, Cswin transformer: A general vision transformer backbone with cross-shaped windows, arXiv preprint arXiv:2107.00652.
[216] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, Z. Liu, End-to-end semi-supervised object detection with soft teacher, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3060–3069.

[217] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, P. Luo, Transtrack: Multiple-object tracking with transformer, arXiv preprint arXiv:2012.15460.
[218] Y. Xu, Y. Ban, G. Delorme, C. Gan, D. Rus, X. Alameda-Pineda, Transcenter: Transformers with dense queries for multiple-object tracking, arXiv preprint arXiv:2103.15145.
[219] F. Zeng, B. Dong, T. Wang, C. Chen, X. Zhang, Y. Wei, Motr: End-to-end multiple-object tracking with transformer, arXiv preprint arXiv:2105.03247.
[220] T. Meinhardt, A. Kirillov, L. Leal-Taixe, C. Feichtenhofer, Trackformer: Multi-object tracking with transformers, arXiv preprint arXiv:2101.02702.
[221] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8741–8750.
[222] Y. Fang, S. Yang, X. Wang, Y. Li, C. Fang, Y. Shan, B. Feng, W. Liu, Instances as queries, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6910–6919.
[223] B. Dong, F. Zeng, T. Wang, X. Zhang, Y. Wei, Solq: Segmenting objects by learning queries, arXiv preprint arXiv:2106.02351.
[224] S. Mehta, M. Rastegari, Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer, in: International Conference on Learning Representations, 2022.

**Xuan-Thuy Vo** received the B.S. degree in electrical and electronic engineering from the University of Science and Technology, the University of Da Nang, Da Nang City, Vietnam, in 2018. He is currently pursuing the Ph. D. degree at the Department of Electrical, Electronic and Computer Engineering, University of Ulsan, South Korea. His current research interests include computer vision and deep learning with a focus on object detection, object segmentation, and multiple object tracking.

**Kang-Hyun Jo** received the Ph.D. degree in computer-controlled machinery from Osaka University, Japan, in 1997. After a year of experience with ETRI as a Post-doctoral Research Fellow, he joined the School of Electrical Engineering, University of Ulsan, Ulsan, Korea where he is currently serving as Faculty Dean. His current research interests include computer vision, robotics, autonomous vehicles, and ambient intelligence. Dr. Jo has served as a Director or an AdCom member with the Institute of Control, Robotics, and Systems (ICROS), the Society of Instrument and Control Engineers (SICE), and IEEE Industrial Electronics Society (IEEE IES) Technical Committee on Human Factors Chair, AdCom member and the Secretary until 2019. At present, he is an Editorial Board Member for international journals, such as the International Journal of Control, Automation, and Systems and the Transactions on Computational Collective Intelligence. He has also been involved in organizing many international conferences such as the International Workshop on Frontiers of Computer Vision, International Conference on Intelligent Computation, International Conference on Industrial Technology, International Conference on Human System Interactions, and Annual Conference of the IEEE Industrial Electronics Society.