 https://doi.org/10.24191/jeesr.v23i1.001

A Review on Object Detection Algorithms based Deep Learning Methods

Wan Xing, Mohd Rizman Sultan Mohd, Juliana Johari, and Fazlina Ahmat Ruslan*

***Abstract*—One of the most dynamic areas in AI research is object detection, a field that continues to evolve due to advancements in chip computing power and deep learning techniques. The central goal of object detection is to identify objects and determine their precise locations by leveraging image processing technology. This application finds utility across diverse industries, such as traffic management, crime scene investigation, and assisted driving. The training process for deep learning-based object identification involves several key steps, thoroughly exploring the data preprocessing, neural network design, prediction, label allocation, and loss calculation. Deep learning-based object detection algorithms can be categorized into three main types: end-to-end algorithms, two-stage algorithms, and one-stage algorithms. Additionally, algorithms can be further divided into anchor-free and anchor-based variants, based on whether bounding boxes are predetermined. This paper begins by reviewing the history and evolution of object detection. It also outlines significant milestones for backbone networks, traditional object detection models, and deep learning-based object detection models, all according to their chronological progression. Furthermore, examples of essential performance evaluation metrics and datasets are provided, while highlighting pressing issues and emerging trends within the field that demand further investigation.**

***Index Terms*— Object detection, deep learning, artificial intelligence, transformer, convolutional network.**

I. INTRODUCTION

OBJECT detection algorithms are used to classify and locate objects in images, calculating their positions with regression functions and judging categories with classification functions. Over the past three decades, there has been a tremendous advancement in object identification algorithms, with a steady transition from theoretical study to practical implementations. They have been widely used in computer-aided design, assisted driving, medical diagnosis, and other fields. There have been roughly two stages in the development of object identification technology: conventional object detection and deep learning-based object detection [1].

This manuscript was submitted on 22nd May 2023 and accepted on 27th August 2023. Wan Xing, Mohd Rizman Sultan Mohd, Juliana Johari, and Fazlina Ahmat Ruslan are from the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia.

*Corresponding author
Email address: fazlina419@uitm.edu.my

Before 2014, most suggested approaches relied on the conventional object detection algorithm. Pre-processing, region selection, feature extraction, and feature classification are among the detection steps and procedures. Pre-processing lessens interference and noise, increasing the variety of images available for training. The standard pre-processing methods are grayscale processing, histogram equalization, median filtering, mean filtering, Gaussian filtering, spatial domain denoising, flipping, and cropping. Traditional object detection algorithms use the technique of sliding windows to scan the entire image and gather all potential positions of objects to find their locations. To find possible candidate locations, feature extraction uses techniques like Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), Haar-like features (Haar), and Local Binary Pattern (LBP). Feature classification methods include Adaptive Boosting (AdaBoost), Support Vector Machines (SVM), Deformable Part Model (DPM), and Decision Tree (DT). However, it is hard to choose appropriate windows when the number of pixels in a picture increases. Due to manually designed operators to extract features, the portability of traditional algorithms is weak.

Deep learning-based object detection algorithms have increasingly risen to the forefront of computer vision research because of the development of large-scale image datasets and high-performance computing chips. The three categories of object identification methods based on deep learning include two-stage object detection models, one-stage object detection models, and end-to-end object detection models. Besides, object detection techniques can be categorized into two groups: anchor-based object identification models and anchor-free object detection models, depending on whether anchor boxes are constructed beforehand. As illustrated in Fig. 1, there are two processes: training and testing.

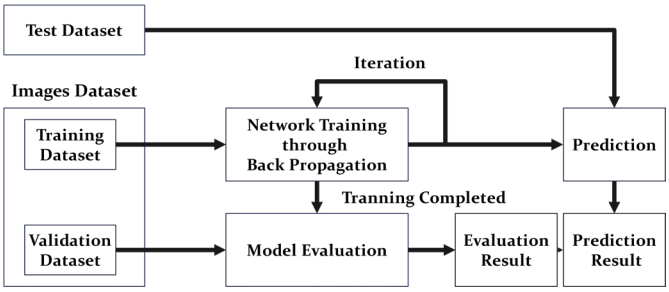

Fig. 1. The procedure for training and testing.

Training involves sending a training dataset into a designed neural network, using a loss function to generate gradients, and updating network parameters through backpropagation. By iterative learning, a neural network that can classify or regress for new unknown data. In the training phase, a validation dataset is included for hyperparameter selection. The training process consists of three specific steps: data pre-processing, network construction and prediction, and label allocation and loss calculation, while testing usually includes only the first two steps. Several frameworks have been created by major corporations like TensorFlow, PyTorch, and PaddlePaddle to assist programmers in swiftly creating object detection algorithms. In summary, deep learning-based object detection is fast evolving in terms of algorithmic research and commercial application.

II. Data Pre-processing

Data augmentation and data normalization are typically included in data pre-processing. Data augmentation comes in two flavors: supervised data augmentation and unsupervised data augmentation. Supervised data augmentation, which includes the single-sample data augmentation and multi-sample data augmentation approaches, is the extension of images using data transformation rules. Information dumping, color modification, and geometric operation are components of single-sample data augmentation. Multiple-sample augmentation methods employ various samples to generate new instances. Geometric transformation contains flipping, rotation, shifting, clipping, deformation, scaling, and other operations. The color transformation includes brightness adjustment, noising, blurring, erasing, filling, etc. Information dropping involves the deliberate removal of a part of a picture, which should avoid excessive deletion and retention of continuous areas. The techniques listed in Table I represent some of the noteworthy approaches.

TABLE I
Units for Magnetic Properties

Data Augmentation	Method Description
Mixup [2]	Directly mix different pictures.
Cutout [3]	Delete a continuous area in an image.
CutMix [4]	Crop a part of one picture and overlay it on another picture.
Manifold Mixup [5]	Mix the input image with the output of middle-hidden layers.
PatchUp [6]	Mix contiguous blocks of features in the hidden space.
SaliencyMix [7]	Add significance analysis.
PuzzleMix [8]	Only clip significant regions and execute some optimization operations.
FMix [9]	Convert the clipping area from a rectangle to an irregular shape.
Co-Mixup [10]	Extract significant regions from multiple samples and mix them.
Mosaic [11]	Splice four pictures by randomly scaling, clipping, and arranging.
Random erasing [12]	Fill an area in the picture with the same pixel value.
Hide-and-Seek [13]	Divide an image into several small patches and delete them randomly.
Grid mask [14]	Remove a region with separate pixel sets.

Unsupervised data augmentation methods include two methods. One is to learn the distribution of a dataset and randomly generate new pictures consistent with the training datasets. The representative algorithm is Generative Adversarial Network (GAN) [15], which produces fake images by training adversarial networks. The other is to learn a data augmentation method suitable for the current task, such as AutoAugment [16]. This algorithm searches for the optimal strategy through reinforcement learning.

In the process of model training, the parameters in the network are constantly updated and transformed, resulting in the distribution transformation, which is called Internal Covariate Shift (ICS). Therefore, its mean and variance need to be transformed to a certain range by data normalization methods to alleviate the vanishing gradient. There are four widely used normalization methods, as shown in Table II.

TABLE II
Four Normalization Methods in Object Detection

Data Norm	Dimension of Normalization
Batch Norm [17]	Batch, width, and height.
Layer Norm [18]	Across all channels, widths, and heights.
Instance Norm [19]	Across width and height.
Group Norm [20]	Within some groups of channels, width, and height.

III. Network Construction and Prediction

The architecture of the object detection network includes the backbone network, feature fusion network, and prediction network, as shown in Fig.2. When inputting the picture into the network, the person in the picture can be detected and enclosed through the bounding box. The core work of object detection is to design an efficient feature extraction network.

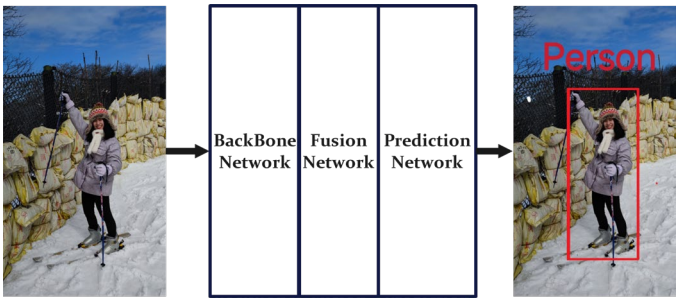

Fig. 2. The network architecture of object detection model.

A. Backbone Network

The backbone is located at the first level of the object detection model, which is usually composed of a convolution network, Full Connected Network (FCN), or transformer modules. Backbones are mainly used to extract features whose complexity determines the time consumption. A representative network is AlexNet [21], which was proposed by Alex Krizhevsky et al. in 2012 and won the championship of the ImageNet competition. VGG Model was proposed in 2014, which further deepens the network depth [22]. Kaiming He et al. proposed ResNet in 2016, which is a significant milestone

and solves the problem of declining accuracy as networks continue to deepen [23]. DenseNet is a more intensive connection method by connecting all layers [24]. DarkNet was proposed in the YOLOv2 framework, which draws on the residual structure in ResNet and uses Batch Normalization (BN) and LeakyReLU [25]. Cross Stage Partial Network (CSPNet) adds more gradient information paths while reducing the amount of computation [26]. The latest research on networks focuses on model re-parameterization techniques, which contain multiple convolution modules in the training stage and merge into one in the inference stage, thus improving the prediction speed. Re-param VGG (RepVGG) adopts high precision multi-branch network learning weights during training and uses low delay single branch network during reasoning [27].

In addition to the conventional backbone networks, recent research hotspots are to build networks based on attention mechanisms. In 2017, the Google team proposed attention mechanisms [28]. In 2020, the DEtection with TRansformers (DETR) built on an attention mechanism was proposed, which is a novel object detection network based on the transformer architecture [29]. Shortly after the proposal of DETR, Deformable DETR was improved to reduce the computation and promote the detection accuracy [30].

To improve their ability to extract features, backbone networks are typically pre-trained on popular datasets like the ImageNet dataset. Transfer learning can utilize parameters that have already been trained using these datasets, which helps to speed up the convergence of model training [31]. Fine-tuning is a popular method for training backbone networks. The model can accelerate convergence by altering the structural elements, such as adjusting the number of output categories or selectively loading the weight parameters.

B. Feature Fusion Network

Feature Fusion Networks (FFN) are based on a multi-scale fusion of features extracted from backbones to improve the object detection capability, especially for detecting small targets submerged in the background. The lower layer features have higher resolution and contain more position information. High-level features have stronger semantic information and a greater receptive field, but their resolution is relatively low, lacking detailed information.

According to the order of fusion and prediction, feature fusion is divided into early fusion and late fusion. In early fusion, the network initially fuses multi-layer features by concatenation or addition also known as skip connection, and then makes predictions using the fused features. In late fusion, several detection results are finally combined based on partially fused layer detection. A Feature Pyramid Network (FPN) is a classical FFN, which up-samples deep-layer information and adds the low-layer information element by element [32]. Multi-scale CNN makes predictions based on multi-scale features and then synthesizes the prediction results [33]. Some other feature fusion networks, such as Path Aggregation Networks (PANet), Multi-level Feature Pyramid Networks (MLFPN), Adaptive Spatial Feature Fusion (ASFF), and Bi-directional Feature Pyramid Networks (BiFPN) are shown in Table III.

TABLE III
SOME FEATURE FUSION METHODS

Methods	Method Description
FPN [34]	Bottom-up and then top-down lateral fusion of features at different scales.
PANet [35]	Add the bottom-up path augmentation and adaptive feature pooling.
MLFPN [36]	Add Thinned U-shape Modules (TUM), Feature Fusion Module (FFM), and Scale-wise Feature Aggregation Module (SFAM) modules.
ASFF [37]	Add learnable feature fusion coefficients.
BiFPN [38]	Integrate top-down and bottom-up to form a module that can be repeatedly stacked and used.

C. Prediction Network

The prediction network also referred to as the detection head, transforms the input feature map dimension produced by the fusion network into the dimension corresponding to the number of object categories and coordinates. Targets in the original image are surrounded by bounding boxes that are drawn based on the predicted coordinates, and categories are annotated. Typically, the regression head and the classification head are two of the parts of the head in YOLOX. [39]. The author of YOLOv7 proposed a strategy for training auxiliary heads that attempts to boost training accuracy without affecting inference time. [40]. The auxiliary heads only appear in the training process and are not used in the prediction process. Based on the Transformer detector, object detection is modeled as an end-to-end bounding box prediction problem, greatly simplifying the complexity of the prediction head.

IV. LABEL ASSIGNMENT AND LOSS CALCULATION

Label assignment primarily concerns how each pixel on the feature map is represented with appropriate learning objectives and how positive and negative samples are assigned. Through backpropagation, the loss, which expresses the discrepancy between the expected outcome and Ground Truth (GT), gives optimizers the gradients.

A. Label Assignment

In object detection, the criteria for label assignment include Intersection over Union (IoU), distance criterion, likelihood estimation, bipartite matching, and Optimal Transport Assignment (OTA). Label assignment methods can also be divided into four types as shown in the following Table IV.

TABLE IV
LABEL ASSINGMENT METHODS IN OBJECT DETECTION

Type	Assignment Criteria	Learning Objective
Anchor box [41]	IoU	Bounding boxes
Anchor free [42]	Gaussian heatmap	Keypoints and radius
key point [43]	IoU	Representative points
Set prediction [29]	Hungarian algorithm	Bounding boxes

IoU is the most used label assignment criterion for determining how much of the entire area of two regions overlap with one another. However, IoU cannot indicate the separation between the anticipated bounding box and the ground truth when they do not intersect. Additionally, it is unable to determine the angle, direction, or aspect ratio. Because the loss function in this situation is not differentiable, neural networks are not optimized. As stated in Table V, some further, improved IoU methods are suggested.

TABLE V
IOU-BASED ALGORITHMS

IoU	Method Description
GIoU [44]	Introduce a penalty when there is no overlap.
DIoU [45]	Add optimization of the distance.
CIoU [45]	Add a parameter of aspect ratio.
EIoU [46]	Calculate length and width separately.
SIoU [47]	Add the angle parameter between the two boxes.

In object detection, there is no exact match between the input image and the label. A picture may contain one or multiple objects. IoU allocates labels according to the local spatial position relationship between predicted anchors and ground truths, which causes many duplicate boxes. There are four rules for judging positive and negative samples using IoU: threshold, Top K boxes, dynamic IoU, and statistical distribution. Some IoU-based label assignment methods are shown in Table VI.

TABLE VI
SOME LABEL ASSIGNMENT METHODS

Method	Description
Guided Anchoring [48]	Create anchors according to conditional distributions (position and scale)
MetaAnchor [49]	Randomly select anchors of any shape
ATSS [50]	Automatic positive and negative sample selection based on statistical aspects of GT
AutoAssign [51]	Fully data-driven assignment
OTA [52]	Find the global optimal partition of the Optimal Transport problem
SimOTA [39]	Faster and simplified OTA

In addition, object detection methods use the distance criterion and assign the corresponding labels based on the distance from the point to the center of the object. Binary matching and likelihood estimation criteria are based on the joint classification and regression loss for the best label assignment. Bipartite graph matching, an end-to-end label assignment technique, is used from a broad perspective to create a set prediction between output and labels [29].

B. Loss Calculation

The loss function can be separated into two categories based on the outcomes of the object detection: regression loss and classification loss. The classification loss assesses the misclassification, whereas the regression loss computes the coordinate errors of the predicted bounding boxes. The weighted sum of the two components represents the total inaccuracy. The object detection network's weights are updated using the backpropagation algorithm by the overall loss. Mean Absolute Error (MAE, L1), Mean Squared Error (MSE, L2), Root Mean Square Error (RMSE), and smooth L1 loss are typical regression loss functions. While L2 calculates the square sum of the distance, L1 calculates the average error of the distance. Smooth L1 loss uses the square function of L2 loss near point zero, which solves the problem that the gradient of L1 loss at point zero is not differentiable, making it smoother and easier to converge [53].

Binary cross-entropy loss, multi-class cross-entropy loss, and focal loss are examples of classification loss functions. A measure of the separation between two probability distributions is called binary cross-entropy, which may be applied to multi-class cross-entropy loss. To address the issue of the excessively imbalanced amount of positive and negative samples in dense object detection tasks, the focal loss function is developed. The learning effect of hard negative samples can be enhanced by this function by including weight parameters based on standard cross-entropy loss. [54].

V. DATASETS AND EVALUATION INDICES

Datasets are essential to the quick development of object detection systems. In general, datasets are split into the training set and the test set. The training set is further subdivided into a training subset and a validation subset. Annotation files from the training set are often in XML, JSON, or YAML formats. Each annotation file contains the category, coordinate, width, and height information of each picture. By testing algorithms on some public datasets, it is convenient to compare the pros and cons of different algorithms through some performance indices.

A. Datasets

Except for some private datasets related to specific tasks, the field of object detection has seen the emergence of numerous well-known datasets over the past ten years, including Open Images, MS COCO, ILSVRC, PASCAL VOC 2007, and PASCAL VOC 2012. These datasets are used by academics, researchers, and engineers to evaluate algorithm performance or for competitions, as shown in Table VII.

TABLE VII
SOME PUBLIC DATASETS IN OBJECT DETECTION

Dataset	Categories	Images	Samples
Pascal VOC 2007 [55]	20	5K	12K
Pascal VOC2012 [56]	20	11K	27K
ILSVRC [57]	200	517K	534K
MS COCO [58]	80	165K	897K
Open Images [59]	6000	1910K	15440K

The Pascal VOC image challenge competition is where Pascal VOC 2007 and Pascal VOC 2012 came from. The standard for assessing the effectiveness of image classification and object recognition algorithms, the ILSVRC dataset is used in the

ImageNet visual challenge from 2010 to 2017. It contains photos of numerous categories of everyday things. The MS COCO dataset contains real-world images that are intended for scene understanding. The largest object detection dataset in the world is called Open Images and is managed by Google. Datasets for several specific application sectors, such as pedestrian identification, face recognition, traffic signal detection, and medical imaging detection, are available in addition to datasets for general object detection.

B. Evaluation Indices

Deep learning-based object detection algorithms come in a variety of forms, making it particularly important to understand how to evaluate an algorithm's quality. The effectiveness of object detection should consider both the accuracy of the forecast and the position of the identified object. Precision (P), Recall (R), Average Precision (AP), mean Average Precision (mAP), and Frames Per Second (FPS) are the key performance evaluation metrics used in the object detection industry.

In the calculations, True Positive (TP) denotes positive samples found by the model, whereas False Positive (FP) denotes negative samples predicted as positive by the model. Similarly, False Negative (FN) labels the positive samples that the model predicts as negative. The P-R curve depicts the overall effectiveness of the detection algorithm. The area under the P-R curve for a specific detected object is referred to as AP. The term "mAP" refers to the average of all categories, while AP only calculates one category. The term "FPS" describes how many photos the model can process per second.

Companies and communities all over the world have made several object detection frameworks available. Among them, the most representative is the Detectron2 launched by Facebook. Detectron2 not only supports milestone object detections, instance segmentation, and pose estimation, but also provides semantic and panoramic segmentation tools. MMDetection is an open-source deep-learning object detection framework jointly developed by SenseTime Technology and the Chinese University of Hong Kong, which is characterized by modular packaging.

VI. Two-stage Object Detection Models

Two-stage object detection models are the earliest object detection algorithms based on deep learning, which have had a profound impact on subsequent algorithm research. The first step focuses on finding positions where objects appear in proposal anchor boxes, and the second step is to classify proposal boxes to find more accurate positions. These algorithms usually have high accuracy but slow speed. The milestones of the two-stage algorithms are shown in Fig. 3.

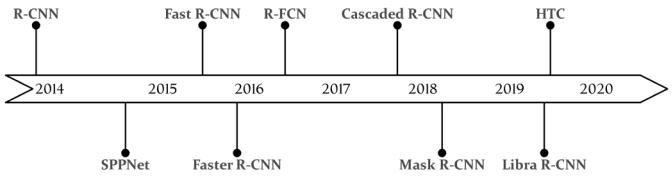

Fig. 3. The milestones of two-stage object detection models.

A. R-CNN

As the first object detection algorithm based on deep learning, Region CNN (R-CNN) was proposed in 2014 [60]. On the VOC2012 dataset, the mAP of this model is 30% higher than the best model before. To train the classifier to identify objects, R-CNN first extracts around 2000 bounding boxes and then feeds each bounding box into a single convolutional neural network for feature extraction. The fundamental principle behind the selective search used by the bounding box selection method is to continually combine related pixels into an entity. This method first separates the image into several little parts, after which it determines how similar any two adjacent regions are. Each step produces areas preserved as bounding boxes as shown in Fig. 4.

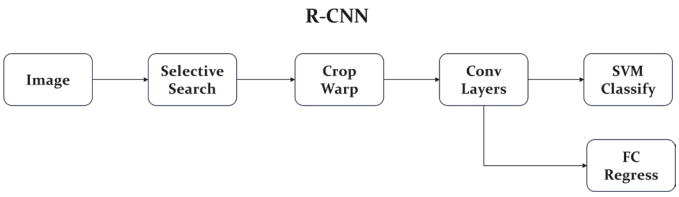

Fig. 4. The network architecture of R-CNN.

B. SPPNet

The Spatial Pyramid Pooling Network (SPPNet) was proposed by Kaiming He et al., which solves the shortcoming of slow speed in R-CNN [61]. The feature map of each candidate region can only be obtained by first entering the original image into the network, which is its most significant enhancement. While guaranteeing that the output is a fixed vector, SPPNet adds Spatial Pyramid Pooling to the CNN structure to allow the network input image to be of any size. In Fig. 5, the network architecture is displayed.

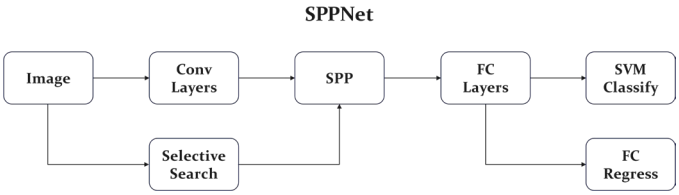

Fig. 5. The network architecture of SPPNet.

C. Fast R-CNN

The authors of R-CNN proposed Fast R-CNN given the shortcomings of their original model [53]. SVM requires additional storage space, resulting in slow training and testing. Another drawback is the R-CNN repeatedly calculates a lot of overleaps. The original model uses a Region of Interest (RoI) pooling layer to output different features to a fixed size. Each batch of 128 RoIs from various pictures is used, which results in an extremely slow training speed. Fast R-CNN minimizes computation by choosing all RoIs from just two images. The Fast R-CNN is shown in Fig. 6.

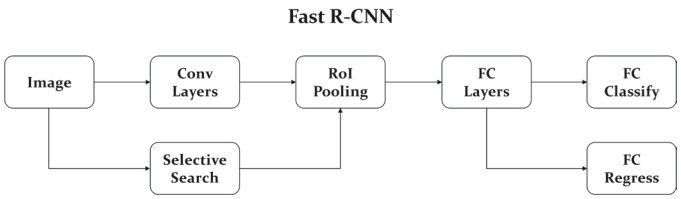

Fig. 6. The network architecture of Fast R-CNN.

D. Faster R-CNN

Ross B. Girshick et al. proposed Faster R-CNN in 2016, further improving the speed and accuracy compared with Fast R-CNN [41]. The Faster R-CNN network is divided into two parts: the shared full convolution network before ROI pooling and the ROI-wise subnet after ROI pooling. In Fast R-CNN, anchor boxes are still generated by the traditional selective search algorithm. Faster R-CNN combines feature extraction, anchor box generation, and coordinate regression in a unified neural network. Besides, a Region Proposal Network (RPN) is proposed to generate anchor boxes. Three aspect ratios and three scale factors are used to produce nine anchors of different sizes. Although there are more than ten thousand anchors, the inference speed is faster because they are fixed. The architecture is shown in Fig. 7.

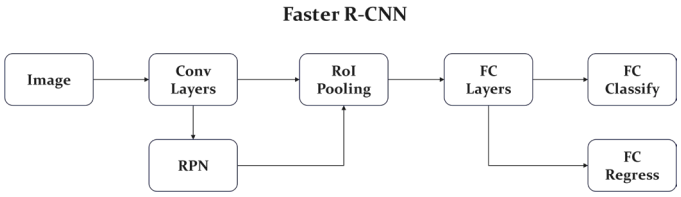

Fig. 7. The network architecture of Faster R-CNN.

E. R-FCN

The main contribution of the Region-based Fully Convolutional Network (R-FCN) is to solve the contradiction between "translation variance in image classification" and "translation variance in object detection" [62]. The model improves the detection speed and accuracy by using position-sensitive score maps. Compared with Faster R-CNN, R-FCN has a deeper shared convolutional network layer, which can obtain more abstract features. There are three branches in R-FCN. The first branch is to perform RPN operations on the feature map to obtain corresponding RoIs. The second branch is to obtain a position-sensitive score map on the feature map for classification. The last branch is to obtain a position-sensitive score map on the feature map for regression as shown in Fig. 8.

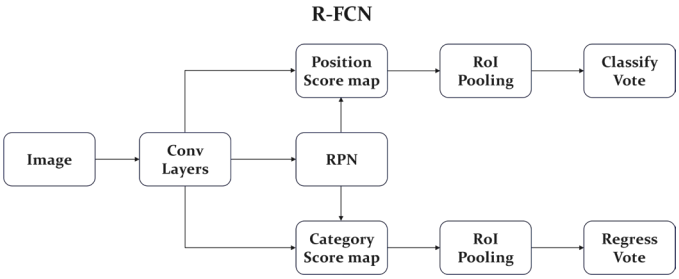

Fig. 8. The network architecture of R-FCN.

F. Other Two-stage Detectors

Mask R-CNN is a combination of Faster R-CNN and FCN, where the former is responsible for object detection and the latter is responsible for object contour [63]. Setting the optimal IoU threshold is often a difficult task, and Zhaowei Cai et al. constructed a Cascade R-CNN by gradually increasing the threshold [64]. To address the imbalance between samples, features, and targets, a new two-stage object detection model, Libra R-CNN, was proposed in 2019 [65]. Another model, Hybrid Task Cascade (HTC) by combining the advantages of Mask R-CNN and Cascaded R-CNN, achieves excellent performance [66]. Due to the complex process involved and relatively slow detection speed, the research gradually shifted to one-stage object detection.

VII. One-stage Object Detection Models

One-stage object detection algorithms do not need to build region proposals and directly generate classification probabilities and bounding box coordinates, which have a faster detection speed. The milestones of one-stage detection models are show in Fig. 9.

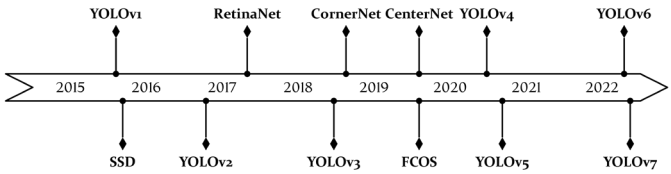

Fig. 9. The milestones of one-stage object detection models.

A. YOLOv1

You Look Only Once (YOLO) is the first one-stage model, which was proposed in 2015 [67]. Before it appeared, object detection algorithms were all two-stage methods. The disadvantage of two-stage methods is that they cannot meet real-time detection. YOLOv1 not only makes classification predictions in the last layer of the neural network but also adds position predictions. It divides an image into grids, and each grid corresponds to the center of N anchor boxes. In the test phase, only the bounding box with the highest probability in the same object category is selected as the prediction result. The prediction speed of YOLOv1 can reach more than 45 FPS, and its shortcoming is that it has low prediction precision on small objects gathered. This is mainly because the grids are sparse and

only two bounding boxes are predicted for each grid. The model is shown in Fig. 10.

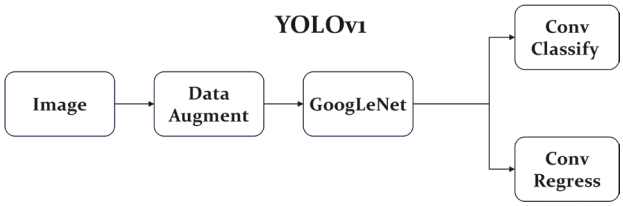

Fig. 10. The network architecture of YOLOv1.

B. YOLOv2

YOLOv2 is an improved version of YOLOv1 by the same author [25]. Firstly, YOLOv2 adds a batch normalization layer behind each convolutional layer, and the dropout layer is no longer used. Secondly, higher image resolution is used for training and an input image size will be randomly selected to enhance the ability to adapt to images of different sizes. Thirdly, YOLOv2 uses more anchor boxes to increase the number of candidate boxes, greatly improving the recall rate. Fourthly, the size of anchor boxes is obtained through clustering analysis based on the K-means algorithm by employing IoU as the distance measurement between two boxes. Lastly, to solve the problem of inaccurate prediction of the bounding box in YOLOv1, YOLOv2 constrains the center of the prediction frame within each fixed grid. The model is shown in Fig. 11.

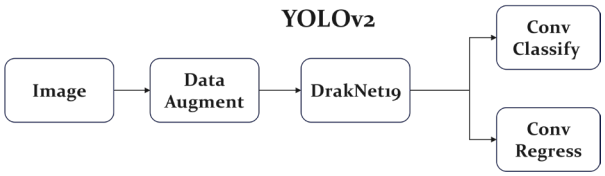

Fig. 11. The network architecture of YOLOv2.

C. YOLOv3

YOLOv3 makes use of three feature maps with different sizes to predict small, medium, and large objects [68]. Small feature maps predict large targets, while large feature maps predict small targets, thus improving accuracy. This model utilizes logistic regression for category prediction because one object can correspond to multiple categories. In addition, YOLOv3 changes the Darknet-19 of YOLOv2 into Darknet-53 and uses the K-means method to obtain nine sizes of anchor boxes. The network is shown in Fig. 12.

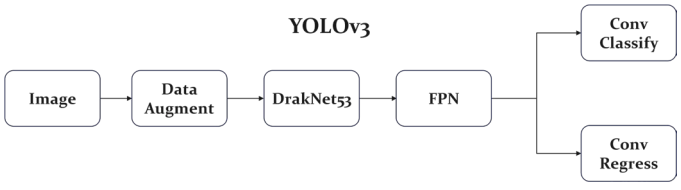

Fig. 12. The network architecture of YOLOv3.

D. YOLOv4

YOLOv4 improves images during training, using Mosaic enhancement, Cross mini-batch Normalization (CmBN), and Self-Adversarial Training (SAT) [11]. This version has a stronger backbone network, CSPDarknet53. In addition, the Mish activation function, label smoothing, and DropBlock were introduced. Besides, YOLOv4 inserted SPP, FPN, and Pyramid Attention Network (PAN) structures into the neck network. Compared with YOLOv3, the regression loss function adopts CIoU Loss, and the Non-Maximum Suppression (NMS) method filtered by the prediction becomes DIoU. The model is shown in Fig. 13.

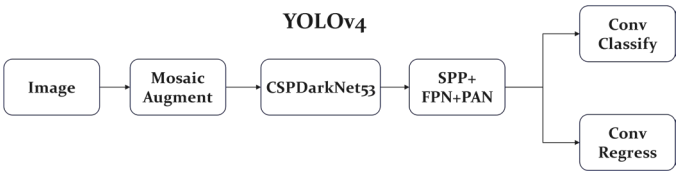

Fig. 13. The network architecture of YOLOv4.

E. YOLOv5

The same Mosaic data enhancement technique used in YOLOv4 is used in YOLOv5. In earlier iterations, anchor boxes had basic length and width values. The initial anchor box value is determined using a different model. The optimal anchor box settings are adaptively determined for each training. To reduce the number of incorrect computations, YOLOv5 adaptively adds the least amount of black edge filling to the original image. Fig. 14 displays the model.

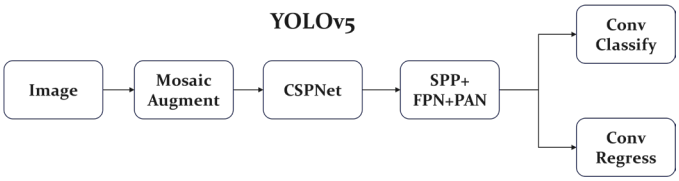

Fig. 14. The network architecture of YOLOv5.

F. YOLOX

YOLOv3 is used as the baseline model by YOLOX, and optimization are made on this basis [39]. YOLOX adds the Exponential Moving Average (EMA) weight update, cosine learning rate mechanism, and other training techniques. It adopts the IoU loss function to train the regression branch, and the Binary Cross Entropy (BCE) loss function to train the classification branch. Besides random horizontal flipping, color Jittering multi-scale data augmentation, and random resized cropping, Mosaic and Mixup data augmentation methods were introduced. YOLOX proposes an end-to-end decoupled head. In label assignment, it uses Simplified OTA (SimOTA) for sample matching as shown in Fig. 15.

Fig. 15. The network architecture of YOLOX.

G. YOLOv6

YOLOv6 has greatly improved the accuracy and detection speed compared with previous models [69]. It not only focuses on the AP and FPS performance but also is very friendly to the industry. For deployment, it provides hardware support for TensorRT, NCNN, OPENVINO, and other platforms. To better be adapted to GPU devices, the Re-parameter (Rep) skill is adopted, and the RepVGG structure is introduced on the backbone to construct EfficientRep. Its Neck also builds Rep-PAN based on Rep and PAN. Like YOLOX, the head is decoupled and has a more efficient structure. YOLOv6 follows the anchor-free approach, abandoning the previous anchor-based approach. Its data augmentation is consistent with YOLOv5, and simOTA is used as a label assignment method as shown in Fig.16.

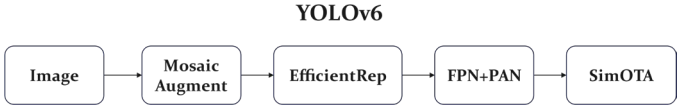

Fig. 16. The network architecture of YOLOv6.

H. YOLOv7

YOLOv7 adopts model re-parameterization techniques to combine multiple calculation modules into one in the reasoning phase [40]. The model is improved based on the Efficient Long-Range Attention Network (ELAN) and Extended ELAN (E-ELAN). In the YOLOv7, a cosine learning rate scheduling strategy is used to adjust the learning rate. The previous YOLO model uses the IoU and GT as soft labels, but YOLOv7 introduces an auxiliary head to assist training. In addition, YOLOv7 also adopts the SimOTA method for sample matching. The model is shown in Fig. 17.

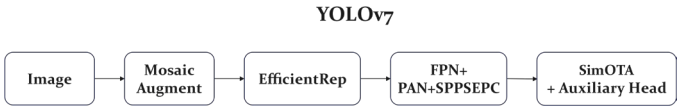

Fig. 17. The network architecture of YOLOv7.

I. SSD

The Single Shot Multibox Detector (SSD) can be viewed as the fusion of YOLO and Faster R-CNN [70]. The detection speed is quite quick since it uses a regression-based model to directly regress the category and position of objects in a network. Additionally, the idea of ROI is also applied. In Fig. 18, the network is displayed.

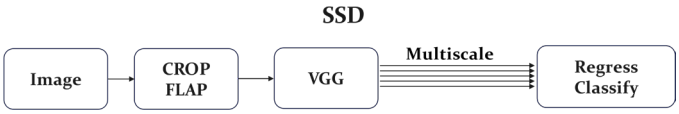

Fig. 18. The network architecture of SSD.

J. Other One-Stage Detectors

Many studies have improved One-stage object detection algorithms. Fully Convolutional One-Stage (FCOS) avoids complex calculations related to anchor boxes during the training process by removing the predefined anchor boxes and predicting them per pixel [71]. Chengjian Feng et al. proposed a Task-aligned One-stage Object Detection (TOOD) method by optimizing object classification and localization [72].

In addition, there is a large amount of research on lightweight object detectors. Jonathan Pedoeem et al. designed a lightweight model YOLO-LITE based on YOLOv2, which can run on both mobile and non-GPU devices [73]. YOLO Nano achieved excellent performance on NVIDIA Jetson Xavier while only occupying approximately 4MB of space [74]. Spiking-YOLO is the first model to use pulse neural networks for object detection, which consumes less energy than lightweight models based on YOLO [75]. SlimYOLOv3 significantly reduces the computational complexity of the model through convolutional layer channel pruning [76].

VIII. The End-to-End Object Detection Models

These models receive original images, and directly output object classification and position. The milestones of end-to-end detection models are shown in Fig. 19.

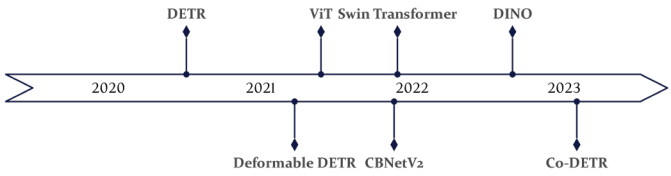

Fig. 19. The milestones of the end-to-end object detection models.

Transformer was proposed by the Google team in 2017 [28]. This model uses the self-attention mechanisms and abandons the RNN structure so that the model can be parallelized and have global information. This is a completely attention-based sequence-to-sequence learning model, which utilizes attention encoders, decoders, and information transfer between them. The encoder of a transformer usually consists of one input layer, several encoding layers, and one output layer, while the decoder is made up of one input layer, several decoding layers, and one output layer. In the input layer, the input sequence is composed of word embedding and position embedding of words, and their sum is taken as the input vector. The network architecture is shown in Fig. 20.

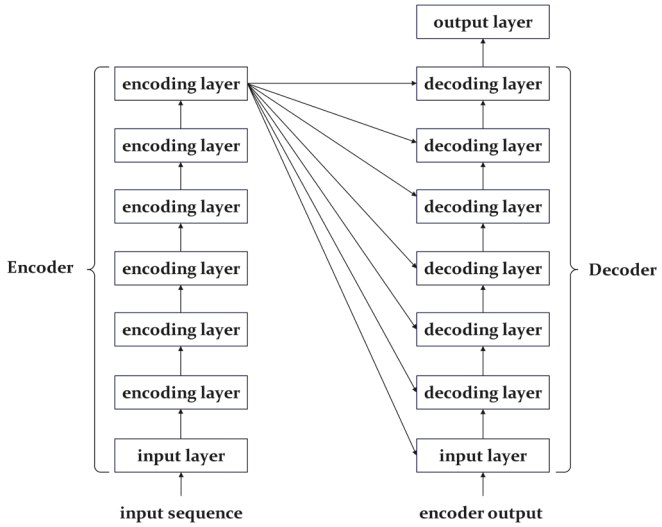

Fig. 20. The network architecture of transformer.

A. DETR

The DETR is the first object detection model based on transformer architecture as shown in Fig. 21 [29]. It removes many hand-designed components, making generating anchors and IoU unnecessary. The Hungarian bipartite graph matching is used to predict objects uniquely, and the object detection problem is transformed into a set global search problem. Because DETR implements an end-to-end object detection mechanism, its object detection results have achieved high performance. Although DETR performs well, there are several issues. First, it needs more training time to converge compared with previous models. Second, performance at detecting small objects is generally not good enough. Wenyu Lv et al. proposed a Real-Time RT-DETR that surpasses YOLO detectors in both accuracy and speed [77].

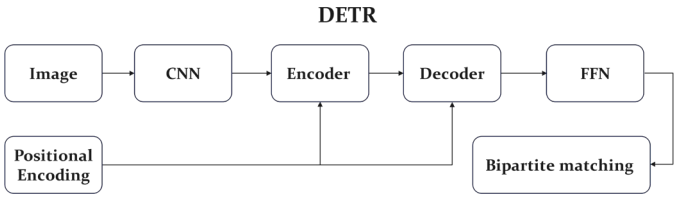

Fig. 21. The network architecture of DETR.

B. Deformable DETR

For DETR, the initialized attention weight assigned to all feature pixels is almost equal. This causes the model to take an extended time to learn and focus on sparse object positions. In the encoder, the computational complexity increases in square order with the number of feature pixels, so it is difficult to process high-resolution features, resulting in poor small object detection. Deformable DETR combines deformable convolution with a transformer to propose a deformable attention mechanism, which solves the problems of slow convergence and high complexity in DETR [78]. Each pixel in this network does not need to be calculated interactively with all feature pixels, but only to be calculated with some pixels obtained from partial sampling.

C. ViT

Vision Transformer (ViT) was proposed by Google in 2020 to apply Transformer to the field of computer vision [79]. ViT directly divides the image into patches with a fixed size and then obtains patch embedding through a linear transformation. When there is enough data for pre-training, ViT outperforms CNN. However, if the training data is not enough, the performance of ViT is generally worse than that of ResNets of the same size. Since attention maps of ViT become similar at a deeper level, Daquan Zhou et al. proposed a Re-attention method to regenerate attention maps [80]. Tokens-to-Token ViT has a 200% reduction in the number of parameters and an improvement in performance on ImageNet [81]. Chun-Fu Chen et al. proposed a dual branch structure, Cross-Attention Multi-Scale Vision Transformer (CrossViT), to combine image patches of different sizes to produce image features [82].

D. Swin Transformer

Swin Transformer was proposed in 2021, which has achieved state-of-the-art performance in many public datasets [83]. Compared with ViT, the Swin Transformer is more efficient and has higher accuracy. The model introduces two key concepts to solve the problems faced by the original ViT hierarchical feature maps and shifted windows. Based on different feature dimensions and the number of layers in each stage, the author designed four different models: Swin-T, Swin-S, Swin-B, and Swin-L.

E. DINO

In response to the unclear meaning of a query and slow model convergence issues in DETR, DAB-DETR reintroduces anchor boxes into DETR to provide query interpretability and accelerate convergence [84]. Based on DAB-DETR, the authors proposed the DN-DETR by using de-noising training to solve the problem of unstable Bipartite graph matching of the DETR decoder, which can accelerate the rate of convergence of the model and significantly improve the detection precision [85]. Furthermore, the authors proposed DETR with Improved deNoising anchoOr boxes (DINO) with the state-of-the-art performance by introducing contrastive de-noising, mixed query selection, and look-forward twice technologies based on the previous two models [86]. Grounding DINO open-set object detector combined DINO with grounded pre-training, which can detect any object with human input such as class names or reference expressions [87].

F. Co-DETR

The authors of Co-DETR point out the one-to-one matching problem of Deformable DETR, which leads to fewer positive queries and inefficient training [88]. The author proposes a simple and effective auxiliary training model Co-DETR, which uses a universal one-to-many matching to improve the training efficiency of encoders and decoders. Co-DETR only adds

auxiliary detection heads during the training phase, so introducing additional computational overhead during the training phase will not affect the efficiency of model inference.

IX. The Development Trends

Object detection has made remarkable achievements in the past twenty years, the performance of models has been continuously improved, and the application fields have been greatly extended. However, there are still some problems that are difficult to solve in this field. This section will discuss the problems faced and future research trends.

A. Lightweight Object Detection

Driven by artificial intelligence and the Internet of Things (IoT), it is urgent to push artificial intelligence to the edge of networks to fully release the potential of big data. Therefore, to ensure a certain accuracy, lightweight models reduce the amount of calculation to speed up the detection. Some large technology companies have introduced edge computing devices with built-in GPU or TPU modules, such as Nvidia Jetson Nano, Google Coral Development Board, and Intel Neurocomputing Stick. MobileNet is based on a streamlined architecture, which uses separable convolutions to build lightweight deep neural networks [89]. ShuffleNet is also a lightweight model, which solves the problem that different convolution output characteristic graphs cannot communicate in MobileNet by adding Pointwise Group Convolution (PGC) [90]. YOLOv7-Tiny conducts a series of ablation experiments on YOLOv7, which runs faster and uses less memory. The model has fewer parameters and is suitable for edge GPU deployment. Lightweight object detection will be one of the main research trends in the future.

B. Multi-task Learning

To realize information sharing and improve the generalization ability of models, researchers introduce multi-task learning into the model to replace single-task learning [91]. Especially, the multi-task learning method based on CNN can realize the convolution sharing of the network structure and improve the generalization ability of models. For multiple computer vision tasks, object detection, segmentation, and image classification are performed simultaneously. More information is obtained, and the performance of individual tasks is greatly improved. However, while maintaining processing speed and improving accuracy, it poses great challenges to researchers.

C. Long-tail Object Detection

Current object detection models are almost trained based on some popular public datasets, such as PASCAL VOC 2007, PASCAL VOC 2012, and MSCOCO. However, the number of target categories of these datasets is limited and far from covering most object categories in the real scene. More importantly, the distribution of objects in the real scene is extremely unbalanced, showing a long tail distribution which is one of the main difficulties that object detection algorithms face [92]. To solve this problem, researchers have constructed a long-tail distributed dataset containing large-scale object types. In addition, some researchers proposed solutions based on sample resampling, loss reweighting, and multi-round training to overcome the problems caused by data imbalance.

D. Transformer-based Object Detection

The Transformer-based object detection models have achieved huge success, which has injected new vitality into the development of this field. The detection performances of these models are better than other object detection models based on traditional convolutional neural networks such as Faster R-CNN, SSD, and YOLO. However, the dense computation, slow convergence, and spatial complexity lead to the low efficiency of transformer-based models. Recent research work has alleviated these problems to a certain extent. Many object detection networks have started using transformers as backbones to replace traditional CNN [93].

E. Self-supervised Learning

Self-supervised learning is a machine learning method that trains models by utilizing automatically generated labels. Tasks from data Self-Supervised Learning are mainly divided into five categories in the field of computer vision, including generative methods, contrastive methods, predictive methods, bootstrapping methods, and regularization methods. Generative methods utilize the generator and discriminator to synthesize objects and labels to train models [94]. Contrastive learning adopts pseudo labels as supervision [95]. Predictive methods provide powerful supervised signals for feature learning and lead to significant improvements in object detection tasks [96]. Bootstrapping methods iteratively train the model by utilizing automatically generated labels and the predicted results of the model, such as Bootstrap Your Own Latent (BYOL) [97]. Regularization methods construct a series of constraints to learn features, such as minimizing redundant features [98].

F. Few-shot and Zero-shot Learning

Object detection is a kind of data-hungry technology because the high accuracy is based on feeding a large amount of data to the model. In real scenes, many tasks need to detect objects, whose categories have never been seen before in the model. This makes conventional training methods no longer applicable. Many tasks do not have so much annotation data, or the cost of acquiring annotation data is very high. Few-shot Learning is the application of Meta-Learning in the field of supervised learning, which learns from a small number of labeled samples [99]. Zero-shot Learning is to solve the problem of unknown object detection [100]. In the training process of the model, these unknown categories are invisible, and there are no relevant labeled training samples. The detection ability in the case of few or zero samples is an important symbol of universality for object detection in the open world [101].

G. Domain Adaptation

Domain adaptation contains data from two different domains [102]. One domain has labeled data as the source data, and unlabeled target data from a new domain. The purpose is to adjust our object detection model so that we can perform well in both domains. Detectors in specific fields can only achieve high detection performance on specified datasets, which have a single application scenario and do not have universality in multiple fields and scenes. By using semi-supervised technology, the performance of domain adaption can be effectively improved, but it deceives the distribution changes to be more biased towards the source data. The Auxiliary Target Domain Oriented Classifier (ATDOC) is introduced as an auxiliary classifier into the target data to reduce classifier bias and improve the quality of pseudo labels [103]. Yunsheng Li et al. proposed a new cross-sample adaptation model that simplifies the alignment between two domains [104].

X. Conclusion

Object detection is an important task of computer vision. With the development of object detection technology for more than 20 years, it has been widely used in industries. However, in the process of its development, there are still challenges, such as extracting effective and multi-scale features, unbalanced positive and negative samples, and label assignment. This paper first introduces the background of object detection technology and then illustrates three parts involved in the process of training and testing, including data preprocessing, network construction, and prediction, as well as label assignment and loss calculation. We also introduce some popular open datasets and performance evaluation indices. Next, one-stage, two-stage, and end-to-end milestone object detection models are introduced by analyzing their advantages and disadvantages. Finally, we made prospects for the development trend of this area. In summary, although the theoretical research on algorithms has made great progress, there is still much room for improvement.

Acknowledgment

The Authors would also like to thank and acknowledge the School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA Shah Alam for their support.

References

[1] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey." arXiv, May 15, 2019. [Online]. http://arxiv.org/abs/1905.05055.

[2] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "M ixup: Beyond Empirical Risk Minimization." arXiv, Apr. 27, 2018.

[3] T. DeVries and G. W. Taylor, "Improved Regularization of Convolutional Neural Networks with Cutout." arXiv, Nov. 29, 2017.

[4] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features." arXiv, Aug. 07, 2019.

[5] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, and Y. Bengio, "Manifold Mixup: Better Representations by Interpolating Hidden States." arXiv, May 11, 2019.

[6] M. Faramarzi, M. Amini, A. Badrinaaraayanan, V. Verma, and S. Chandar, "PatchUp: A Regularization Technique for Convolutional Neural Networks." arXiv, Jun. 14, 2020.

[7] A. F. M. S. Uddin, M. S. Monira, W. Shin, T. Chung, and S.-H. Bae, "SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization." arXiv, Jul. 27, 2021.

[8] J.-H. Kim, W. Choo, and H. O. Song, "Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup." arXiv, Dec. 30, 2020.

[9] E. Harris, A. Marcu, M. Painter, M. Niranjan, A. Prügel-Bennett, and J. Hare, "FMix: Enhancing Mixed Sample Data Augmentation." arXiv, Feb. 28, 2021.

[10] J.-H. Kim, W. Choo, H. Jeong, and H. O. Song, "Co-Mixup: Saliency Guided Joint Mixup with Supermodular Diversity." arXiv, Feb. 05, 2021.

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

[12] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation." arXiv, Nov. 16, 2017.

[13] K. K. Singh, H. Yu, A. Sarmasi, G. Pradeep, and Y. J. Lee, "Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond." arXiv, Nov. 06, 2018.

[14] P. Chen, S. Liu, H. Zhao, and J. Jia, "GridMask Data Augmentation." arXiv, Jan. 13, 2020.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Advances in Neural Information Processing Systems, 2014, vol. 27. [Online]. https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

[16] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning Augmentation Policies from Data." arXiv, Apr. 11, 2019. [Online]. http://arxiv.org/abs/1805.09501.

[17] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." arXiv, Mar. 02, 2015. [Online]. http://arxiv.org/abs/1502.03167.

[18] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization." arXiv, Jul. 21, 2016.

[19] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance Normalization: The Missing Ingredient for Fast Stylization." arXiv, Nov. 06, 2017.

[20] Y. Wu and K. He, "Group Normalization." arXiv, Jun. 11, 2018.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun.2017.

[22] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, Apr. 10, 2015.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770–778.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 2261–2269.

[25] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger." arXiv, Dec. 25, 2016. [Online]. http://arxiv.org/abs/1612.08242.

[26] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN." arXiv, Nov. 26, 2019. [Online]. http://arxiv.org/abs/1911.11929.

[27] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets Great Again." arXiv, Mar. 29, 2021.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need." arXiv, Dec. 05, 2017.

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

[30] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection." arXiv, Mar. 17, 2021.

[31] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, "Model Adaptation: Unsupervised Domain Adaptation Without Source Data," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020, pp. 9638–9647.

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection." arXiv, Apr. 19, 2017. [Online]. http://arxiv.org/abs/1612.03144.

[33] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection." arXiv, Jul. 25, 2016. [Online]. http://arxiv.org/abs/1607.07155.
[34] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, Jul. 2017, pp. 936–944.
[35] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, Jun. 2018, pp. 8759–8768.
[36] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network." arXiv, Jan. 06, 2019. [Online]. http://arxiv.org/abs/1811.04533.
[37] S. Liu, D. Huang, and Y. Wang, "Learning Spatial Fusion for Single-Shot Object Detection." arXiv, Nov. 24, 2019.
[38] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection." arXiv, Jul. 27, 2020.
[39] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021." arXiv, Aug. 05, 2021.
[40] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." arXiv, Jul. 06, 2022.
[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun.2017.
[42] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection." arXiv, Apr. 18, 2019.
[43] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," in Computer Vision – ECCV 2018, Cham, 2018, pp. 765–781.
[44] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression." arXiv, Apr. 14, 2019. [Online]. http://arxiv.org/abs/1902.09630.
[45] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression." arXiv, Nov. 19, 2019.
[46] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and Efficient IoU Loss for Accurate Bounding Box Regression." arXiv, Jul. 15, 2022.
[47] Z. Gevorgyan, "SIoU Loss: More Powerful Learning for Bounding Box Regression." arXiv, May 25, 2022.
[48] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region Proposal by Guided Anchoring." arXiv, Apr. 12, 2019.
[49] T. Yang, X. Zhang, Z. Li, W. Zhang, and J. Sun, "MetaAnchor: Learning to Detect Objects with Customized Anchors." arXiv, Nov. 06, 2018.
[50] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection." arXiv, Jun. 20, 2020.
[51] B. Zhu, J. Wang, Z. Jiang, F. Zong, S. Liu, Z. Li, and J. Sun, "AutoAssign: Differentiable Label Assignment for Dense Object Detection." arXiv, Nov. 25, 2020.
[52] Z. Ge, S. Liu, Z. Li, O. Yoshie, and J. Sun, "OTA: Optimal Transport Assignment for Object Detection." arXiv, Mar. 26, 2021.
[53] R. Girshick, "Fast R-CNN," in 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.
[54] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection." arXiv, Feb. 07, 2018. [Online]. http://arxiv.org/abs/1708.02002.
[55] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results." [Online]. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.
[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." [Online]. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
[57] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge." arXiv, Jan. 29, 2015.
[58] "COCO Dataset." Microsoft Company, 2014. [Online]. https://cocodataset.org/#home.
[59] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale," IJCV, 2020.
[60] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation." arXiv, Oct. 22, 2014. [Online]. http://arxiv.org/abs/1311.2524.
[61] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," vol. 8691, 2014, pp. 346–361.
[62] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks." arXiv, Jun. 21, 2016.
[63] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN." arXiv, Jan. 24, 2018.
[64] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, Jun. 2018, pp. 6154–6162.
[65] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection." arXiv, Apr. 04, 2019. [Online]. http://arxiv.org/abs/1904.02701.
[66] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid Task Cascade for Instance Segmentation." arXiv, Apr. 09, 2019. [Online]. http://arxiv.org/abs/1901.07518.
[67] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 779–788.
[68] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement." arXiv, Apr. 08, 2018.
[69] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications." arXiv, Sep. 07, 2022.
[70] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in Computer Vision – ECCV 2016, Cham, 2016, pp. 21–37.
[71] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully Convolutional One-Stage Object Detection." arXiv, Aug. 20, 2019.
[72] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: Task-aligned One-stage Object Detection." arXiv, Aug. 28, 2021.
[73] J. Pedoeem and R. Huang, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers." arXiv, Nov. 13, 2018.
[74] A. Wong, M. Famuori, M. J. Shafiee, F. Li, B. Chwyl, and J. Chung, "YOLO Nano: a Highly Compact You Only Look Once Convolutional Neural Network for Object Detection." arXiv, Oct. 02, 2019.
[75] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking Neural Network for Energy-Efficient Object Detection." arXiv, Nov. 24, 2019.
[76] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Oct. 2019, pp. 37–45.
[77] W. Lv, Y. Zhao, S. Xu, J. Wei, G. Wang, C. Cui, Y. Du, Q. Dang, and Y. Liu, "DETRs Beat YOLOs on Real-time Object Detection." arXiv, Jul. 06, 2023.
[78] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable Convolutional Networks," p. 10.
[79] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, Jun. 03, 2021.
[80] D. Zhou, B. Kang, X. Jin, L. Yang, X. Lian, Z. Jiang, Q. Hou, and J. Feng, "DeepViT: Towards Deeper Vision Transformer." arXiv, Apr. 19, 2021.
[81] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet." arXiv, Nov. 30, 2021.
[82] C.-F. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification." arXiv, Aug. 22, 2021.
[83] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows." arXiv, Aug. 17, 2021.
[84] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR." arXiv, Mar. 30, 2022.

[85] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR Training by Introducing Query DeNoising." arXiv, Dec. 08, 2022.
[86] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection." arXiv, Jul. 11, 2022.
[87] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection." arXiv, Mar. 20, 2023.
[88] Z. Zong, G. Song, and Y. Liu, "DETRs with Collaborative Hybrid Assignments Training." arXiv, Jul. 02, 2023.
[89] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, Apr. 16, 2017. [Online]. http://arxiv.org/abs/1704.04861.
[90] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018, pp. 6848–6856.
[91] A. Khattar, S. Hegde, and R. Hebbalaguppe, "Cross-Domain Multi-task Learning for Object Detection and Saliency Estimation," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Jun. 2021, pp. 3634–3643.
[92] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep Long-Tailed Learning: A Survey." arXiv, Apr. 15, 2023.
[93] T. Shehzadi, K. A. Hashmi, D. Stricker, and M. Z. Afzal, "Object Detection with Transformers: A Review." arXiv, Jul. 10, 2023.
[94] Y. Lu, M. Shen, H. Wang, and W. Wei, "Machine Learning for Synthetic Data Generation: A Review." arXiv, Jun. 16, 2023.
[95] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A Survey on Contrastive Self-supervised Learning." arXiv, Feb. 07, 2021.
[96] [96] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations." arXiv, Mar. 20, 2018.
[97] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised Learning." arXiv, Sep. 10, 2020.
[98] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow Twins: Self-Supervised Learning via Redundancy Reduction." arXiv, Jun. 14, 2021.
[99] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a Few Examples: A Survey on Few-Shot Learning." arXiv, Mar. 29, 2020.
[100] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A Survey of Zero-Shot Learning: Settings, Methods, and Applications," ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, p. 13:1-13:37, 2019.
[101] [K. J. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards Open World Object Detection." arXiv, May 09, 2021. [Online]. http://arxiv.org/abs/2103.02603.
[102] M. Wang and W. Deng, "Deep Visual Domain Adaptation: A Survey," Neurocomputing, vol. 312, pp. 135–153, Oct.2018.
[103] J. Liang, D. Hu, and J. Feng, "Domain Adaptation with Auxiliary Target Domain-Oriented Classifier." arXiv, Dec. 15, 2021.
[104] Y. Li, L. Yuan, Y. Chen, P. Wang, and N. Vasconcelos, "Dynamic Transfer for Multi-Source Domain Adaptation." arXiv, Mar. 18, 2021.