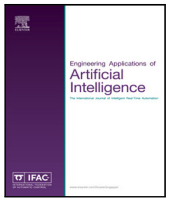
Engineering Applications of Artificial Intelligence 133 (2024) 108458

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Survey paper

The evolution of object detection methods

Yibo Sun [a,*], Zhe Sun [b], Weitong Chen [a]

[a] The University of Adelaide, Adelaide, 5000, South Australia, Australia
[b] Broadband Wireless Communication Technology Engineering Research Center of the Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China

ARTICLE INFO

Keywords:
Object detection
Transformer
Convolution neural network
Deep learning

ABSTRACT

Object detection is one of the most important domains in computer vision tasks, which is an important branch of artificial intelligence. It aims at finding and locating the accurate position of objects in given pictures or videos. With the development of deep learning techniques, more powerful and robust algorithms have emerged to deal with multi-scale, high-level features to overcome the limitations of traditional pipeline of object detectors. The popularity of transformer framework enables larger capacity datasets by processing self-attention mechanism, and the object detection methods have evolved into a new era. This paper first reviews traditional object detection pipeline and brief history of deep learning, afterwards it focuses on the classification of deep learning-based object detection methods covering Convolution Neural Network based and transformer-based methods. Commonly used datasets and metrics are also covered in the next part. The Convolution Neural Network based methods mainly contain two-stage and one-stage detectors, Convolution Neural Network is the underlying structure of these methods convolutional stages are fundamental parts. Transformer-based models convert traditional object detection issues into end-to-end detection, which is widely used in dealing with images. Finally, the promising future of object detection areas are listed to show guidance on future work.

1. Introduction

Human vision plays a crucial role in collecting information from the external world. Images and videos are the main sources of visual information, which have been used widely in computer vision areas, and object detection is one of the branches of computer vision. Object detection aims at identifying and localizing the specific object in images or videos, and has been widespreadly used in many areas such as video surveillance. Ingle and Kim (2022), Ma et al. (2021), face detection (Chen and Joo, 2021; Qi et al., 2022), autonomous driving (Hu et al., 2023b; Burnett et al., 2021), etc. Traditional object detection consists of three major steps (Zhao et al., 2019): (1) Scan the input image with sliding windows for more target objects; (2) Extract some semantic features by selecting extraction methods; (3) Use selected features and classifiers to do classifications.

Traditional object detection methods primarily utilize sliding windows to select candidate regions, it is hard to find the satisfied number of sliding windows. The Scale-Invariant Feature Transform (SIFT) (Burger and Burge, 2022) method focuses on detecting key points in images and generating local descriptors around the selected key points. The selected key points are mainly based on image gradient, Gaussian difference pyramid, etc. to ensure the stability in different scales and rotations. The histogram of oriented gradients(HOG) (Dalal and Triggs, 2005) descriptor calculates the gradient and corresponding gradient histogram on the dense grid for image cells. By combining multiple cells into blocks and calculating feature vector for each block, the method applies a fixed-size sliding window in classification. However, traditional object detection methods mainly rely on manually designed features, which struggle to remain accuracy and robustness with diverse objects and complex background. Furthermore, manually designed features are not universal among various domains, resulting in low accuracy for common models and high time complexity in region selection.

In 2006, Hinton and Salakhutdinov (2006) proposed the solution of reducing the dimensionality of high-dimension data in the neural network models. This is a breakthrough in academic area, at the same time, the performance of high-performance computer systems has developed rapidly, such as GPU. In 2012, Hinton and his research group won the champion in the competition of ImageNet Image Recognition Competition (Krizhevsky et al., 2012) with the AlexNet (Hinton et al., 2012) deep learning model. The activation function of AlexNet utilized the rectified linear unit(RELU) (Agarap, 2018) function which has completely solved the problem of gradient disappearance. At the same time, similar network structures has prompted, such as Residual Net(ResNet) (He et al., 2016), Visual Geometry Group(VGG) (Rani

* Corresponding author.
E-mail address: yibo.sun@adelaide.edu.au (Y. Sun).

https://doi.org/10.1016/j.engappai.2024.108458
Received 2 November 2023; Received in revised form 1 April 2024; Accepted 14 April 2024
Available online 23 April 2024

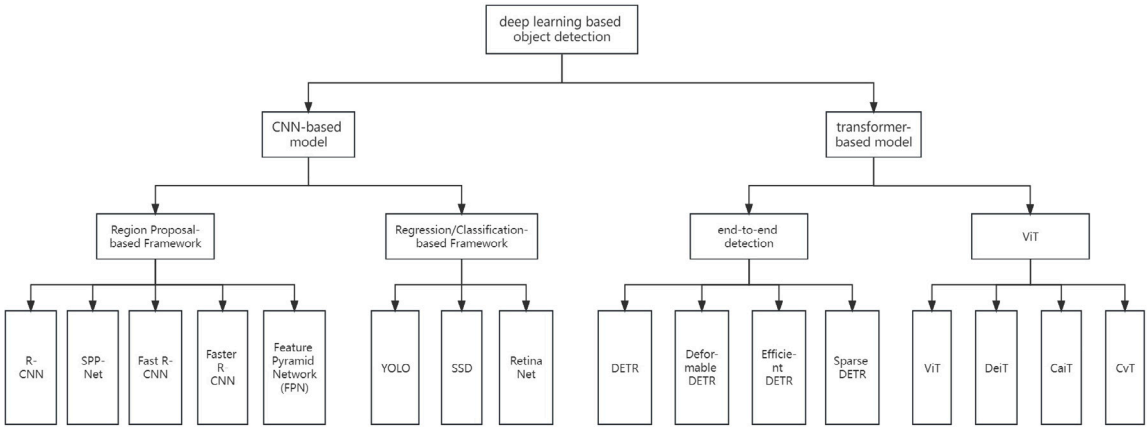


Fig. 1. Classification diagram of deep learning-based object detection.

et al., 2023), ZFNet (Zeiler and Fergus, 2014) and so on. The appearance of these networks are evolutionary in neural networks, with the utilization of batch normalization (Ioffe and Szegedy, 2015) the training efficiency has significantly improved.

With the booming of deep learning technology, deep learning-based object detection have surpassed traditional methods. Deep learning-based methods employ intricate hierarchical architectures, enabling them to train more sophisticated semantic features than traditional object detection methods. Furthermore, convolutional neural networks(CNNs) exhibit formidable learning capabilities, allowing them to represent the features more effectively with larger datasets than traditional object detection methods. Consequently, the robustness of deep learning-based methods is significantly improved, eliminating the necessity for manually designed, hand-crafted features.

This survey focuses on reviewing and analyzing the mainstream on deep learning-based object detection into Convolution Neural Network(CNN) based and transformer-based methods. Existing surveys conclude the development of major branches of methods in object detection area, but may ignore the growing trend of some solutions and novel application areas as the fast development of the artificial intelligence background. This paper tries to convey more thoroughly around the history and future expectations in object detection related to technical points and applications.

(1) This paper provides a comprehensive view of technical issues and applications on deep-learning based object detection area, which is generally divided into Convolution Neural Network(CNN) based methods and transformer based methods.

(2) By reviewing the general history of neural network development, this paper illustrates the growing trend of object detection areas, major topics in the same area. It shows guidance for readers to learn about the major research trends on object detection.

(3) Detailed descriptions in many aspects are presented in this paper, by showing the terminologies and important concepts and comparing the performance results in different methods.

Above all are the contributions among our survey paper, our survey takes reference from Jiao et al. (2019) and Wu et al. (2020), which reviewed the deep learning-based object detection results but ignored the growing trend of self-attention mechanism in transformer models. Zou et al. (2023) reviewed the history of object detection thoroughly and comprehensively, but the detailed terminologies have not been covered smoothly and specialized.

2. Classification of deep learning-based object detection methods

Deep learning-based object detection methods has becoming the mainstream in object detection researches. These models can be divided into two types: CNN-based models and transformer-based models. The classification of algorithms is shown in Fig. 1.

2.1. General structure of convolutional neural network(CNN)

At present, ResNet and VGG are the most popular architectures in CNN networks. Each network contains three types of layers: convolutional layer, pooling layer and fully-connected(FC) layer. The input of layer into network consists of several 3D pixels, which represent the dimensions of pictures (e.g. width, length, height). The feature detector is also required in the convolutional step, which moves around the receptive fields of images to extract useful features. A filter is then applied to calculate the value of features in the scanned area, and moved with a stride, until the region of all image has been swept. The result of the filter is known as feature map, and the activation function Relu (Hinton and Salakhutdinov, 2006; Agarap, 2018) is utilized to increase the nonlinearity of the model. The pooling layer aims at reducing the dimension of input features, which is also known as downsampling (Zhang et al., 2011), in order to make the selected features more robust and filtering out some meaningless information. Max pooling and average pooling serves as the function of reducing dimensions.

After filtering and pooling in these layers, the pixels are sent into the fully connected layers so that the output neurons can be transferred into binary values with softmax (Chen et al., 2022) classification layer. For example, the VGG network consists of a total of 13 convolutional layers, 3 fully connected (FC) layers, 3 max-pooling layers, and a softmax classification layer. The convolutional feature maps are generated through the convolution of 3 * 3 filter windows, and the feature map resolutions are reduced using max-pooling layers with a stride of 2. The ResNet101 (Li et al., 2018b) has the structure of 5 convolutional stages and a total of 101 convolutional layers.

Based on the hierarchical and convolutional structure of CNN (Kavukcuoglu et al., 2010), which is capable of learning the features of complex datasets more easily. It is more accurate and convenient to automatically learn from the complex backgrounds, and the semantic features can be more robust after several convolutional steps. Since the CNN network has higher learning capacity and a deeper structure, it is more suitable for handling challenges in high-dimensional datasets. Therefore, CNN is popular among computer vision problems especially in object detection methods.

CNN-based models leverage the hierarchical structure of CNN (Kavukcuoglu et al., 2010), with higher learning capacity and computational performance, rendering them better suited for adapting to complex datasets in comparison with traditional object detection methods. These CNN-based methods can be divided into two types: (1) candidate region-based model(two-stage detector); (2) classification/regression-based model(one-stage detector). The candidate region-based model is a two-stage type method, which generates the region of interest(RoI) first to select the candidate bounding boxes, then it extracts the

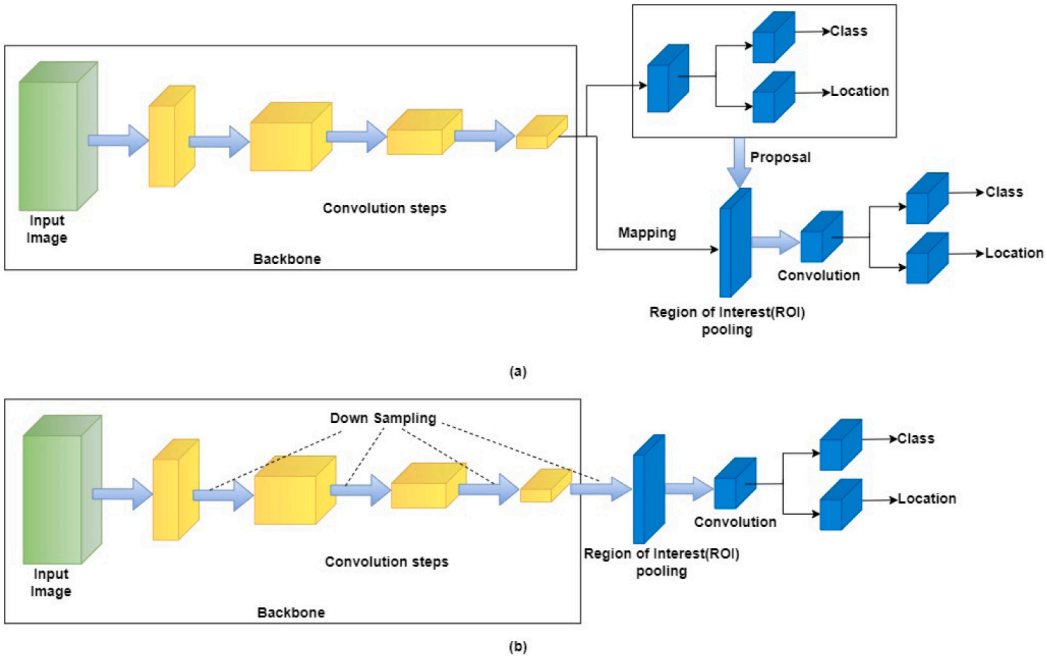


Fig. 2. The comparison between two types of object detection, part (a) shows the basic structure of two-stage detector and part (b) shows the structure of one-stage type.

features from the candidate boxes by using RoI pooling layer to find the category of objects. The R-CNN (Girshick et al., 2014) series methods are typical types of candidate region-based models. In contrast, the regression/classification-based framework just match the objects from original images to bounding boxes without the need of candidate box extraction. The YOLO (Redmon et al., 2016) series are typical types of classification/regression based models. Fig. 2 clearly shows the differences between two types of object detection frameworks.

2.2. General structure of transformer

In recent years, Transformer structure has become popular in computer vision. In traditional sequence-to-sequence model, long-distance dependencies are hard to capture because the model needs to process the input sequence sequentially, leading difficulties in gradient propagations. In 2015, Vaswani et al. (2017) proposed the self-attention based model Transformer. The core idea of self-attention aims at computing a weight matrix by combining the weighted representation information on different positions of input sequences to get output representation for each position. The weight matrix can be dynamically modified according to different features among input sequences so as to adapt to various tasks. By utilizing the self-attention mechanism, the model is able to pay more attention to the positions of sequences so that the training steps can be more robust.

Transformers utilize the self-attention mechanism to establish global dependencies between different points in sequence (Khan et al., 2022), which allows the model to dynamically compute the attentional weights between each position among others for long-distance dependencies. The self-attention is an essential part of Transformers, which simulates the interactions between entities in sequences. The sequence of n entities $(x_1, x_2, \ldots, x_n)$ can be defined as$(X \in \mathbb{R}^{n\times d})$ where d represents the dimension of each entity in sequence. For each input entity, the mechanism replicates input sequence three times, where learnable matrixes can be served as Query$(W^Q \in \mathbb{R}^{d\times d_q})$,Key$(W^K \in \mathbb{R}^{d\times d_k})$, and Value$(W^V \in \mathbb{R}^{d\times d_v})$, where $d_q = d_k$. These learnable matrixes are mainly used to calculate attention weights by projecting input sequences on weight matrixes for $Q = XW^Q$, $K = XW^K$, $V = XW^V$. The self-attention weight can be computed as $Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right)V$.

Furthermore, attention-based mechanism avoids the loop and convolution stages in traditional sequence-to-sequence model, so that the model can be effectively calculated in parallel thus accelerating the training process greatly. The multi-head attention module, containing several self-attention modules, aims at paying more attention to differences between elements in the sequence. Finally, the position encoding is employed to maintain the sequence of words.

Self-attention mechanism is the fundamental part of Transformer, its main structure is based on encoder–decoder structure. The encoders process input sequences, along with several identical parts where multi-head self-attention layer and feedforward neural network layer are the core components. The output sequences of encoders support for decoders. Positional encoding is employed to label positional information to distinguish words or tokens on separate positions, and the residual connection layer is used to help alleviate problem of gradient loss. The decoders have similar steps as encoders, they process output encoding layers. Decoders have six identical blocks, similar to encoders, multi-head self-attention layer and feedforward neural network layer process the sequences of information, while the third sublayer performs the multi-head attention on the output of encoder. The general Transformer structures contain 6-encoder and 6-decoder, which is commonly used in subtasks.

The Transformer structure has made great breakthrough in Nature Language Processing(NLP) area, and major contributions are highly related to NLP. The NLP has taken a major position in transformer application, it was not until the appearance of DETR (Carion et al., 2020) and Vision Transformer (Dosovitskiy et al., 2020) model the transformer has been popularized among object detection and other computer vision tasks. The major achievements are listed in Table 1.

Transformer-based object detection models are based on the self-attention mechanism to establish global dependencies between different points in sequence (Vaswani et al., 2017; Khan et al., 2022), which allows model to dynamically compute the attentional weights between positions. Transformer-based methods can be divided into two main categories: (1) DETR (Carion et al., 2020) series model; (2) ViT (Dosovitskiy et al., 2020) series model. The DETR series model simplifies the object detection into an end-to-end framework by using encoder–decoder module of transformer. Comparing with CNN-based model, DETR removes the Non-maximum suppression(NMS) module

Table 1
Major contributions on papers related to Transformer.

Papers	Contributions
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2018)	Using bidirectional learning pre-trained method in transformer by taking into account the contextual information of sentences to make the model more powerful in understanding the linguistic context. The BERT model is its fine-tunability. The pre-training part of the model can be fine-tuned on a variety of NLP tasks in order to transfer the learned linguistic knowledge to a specific task.
Improving Language Understanding by Generative Pre-training (Radford et al., 2018) Language Models are Unsupervised Multitask Learners (Radford et al., 2019) Language Models are Few-Shot Learners (Brown et al., 2020)	GPT proposes a pre-training and then fine-tuning framework, where models learn rich linguistic knowledge by pre-training on large-scale text corpus and then adapt to specific tasks by fine-tuning. GPT-3 achieves excellent performance on a wide range of natural language processing tasks without the need for large-scale task-specific fine-tuning.
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (Raffel et al., 2020)	The model proposes a generic text-to-text framework that unifies various natural language processing tasks in the form of text input and text output. The model is pre-trained by using a large-scale text corpus, learning a wide range of language knowledge and task knowledge. The potential of combining multiple natural language processing tasks for training together is demonstrated, resulting in improved model efficiency and performance.
End-to-end object detection with transformers (Carion et al., 2020)	The DETR model in this paper introduces the Transformer architecture to the field of target detection for the first time and proposes an end-to-end target detection method. DETR abandons the need for predefined anchor boxes in traditional target detection and instead directly outputs the target's position information through the attention mechanism. It reduces the complexity of adjusting and designing anchor points and improves the flexibility of the model.
An image is worth 16 × 16 words: Transformers for image recognition at scale (Dosovitskiy et al., 2020)	The ViT model in this paper has successfully utilized the transformer structure in image classification tasks. It segments an image into a set of small image chunks and uses Transformer's attention mechanism to capture global and local image information. The ViT model performs well in processing images of different resolutions, which means ViT has potential in a wide range of visual tasks, not limited to image classification.

which is the hand-crafted process to deal with redundant boxes. These types of models are widely used in detecting tasks. Meanwhile, the ViT series model divides the whole images into patches, where each path contains a piece of the image area. Then, the pixel values in each path are stacked into a vector and use these vectors as input to the model. These types of models are widely used in image classification tasks.

This section has mainly discussed the basic structures of the CNN model and the Transformer model. In the following two sections, the paper will discuss the major evolution of methods related to these categories.

3. CNN-based models

The CNN-based object detection can be mainly divided into two major parts: two-stage object detection and one-stage object detection. The two-stage models mainly use region of interest(RoI) to generate candidate bounding boxes and then extract features from bounding boxes to find objects. In comparison, the one-stage methods do not use RoI mechanism to locate objects, both steps are combined in just one stage.

3.1. Two-stage object detection

3.1.1. R-CNN

R-CNN is the typical structure of a region proposal-based framework. In 2014, Girshick et al. (2014) proposed the R-CNN method which successfully combined selective search (Uijlings et al., 2013) with the CNN network. Their work shows a groundbreaking work that the CNN could increase the performance obviously in PASCAL (Everingham et al., 2010) datasets than previous detection methods. This milestone clearly represents that deep-learning-based methods are becoming mainstream in the object detection area.

R-CNN model consists of four major modules. The first module generates candidate region proposals, employing the selective search method to identify detection areas, which include the target areas in each region. The second module aims to extract a 4096-dimensional feature vector in each region proposal, with each region in a fixed size at 227 * 227 pixels (Zeiler and Fergus, 2014). This model consists of five convolutional layers and two fully connected layers in support of feature extraction. The third module inputs the extracted features into a classifier, with Support Vector Machine(SVM) (Cervantes et al., 2020) serving as the classifier in this model to determine whether the selected region contains target and which target belongs to. To ensure the quality of the selected region, the non-maximum suppression(NMS) (Shepley et al., 2023) is employed in this model to filter out those less important features. The last module is the bounding box regression module, which aims to ensure that the target object can be located more accurately. The general procedures of R-CNN is shown in Fig. 3.

R-CNN has made significant advancements in the field of object detection. The introduction of region proposal enables the algorithm to conduct feature extraction and classification solely on the candidate regions without the need to slide through all the images. This substantially reduces computational overhead and enhances efficiency. Furthermore, R-CNN utilizes the AlexNet model for feature extraction, enabling the algorithm to learn richer and more abstract feature representations from images, consequently improving the detection accuracy.

Although R-CNN has made great breakthrough in object detection, it still has some limitations in application areas:

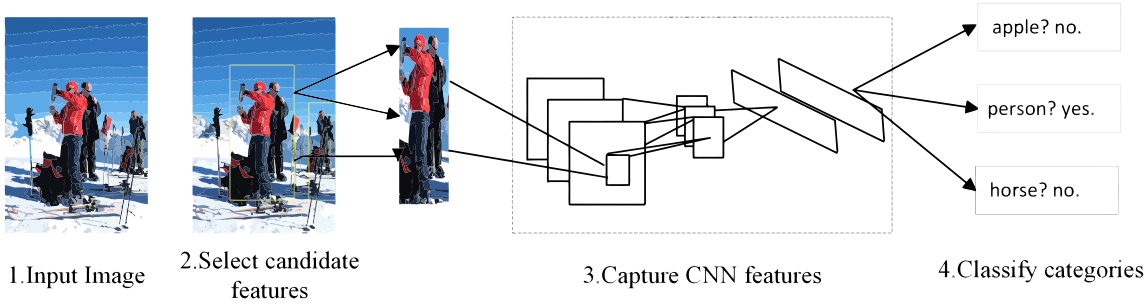


Fig. 3. General structure of R-CNN.

(1) The model largely maintains the CNN structure, requiring the input of images to be a fixed size(227 * 227 pixels), it takes too much time in preprocessing images when the input size is unbalanced.

(2) Concerning the overlap between each candidate boxes, adjacent regions often share similar features, resulting in redundant calculation and decreased efficiency on similar region proposals, which are selected for feature extraction.

(3) R-CNN involves a multi-stage process, which can be time-consuming as it requires separate training for candidate region generation, feature extraction, and target classification.

(4) The precision of bounding box definitions may suffer when using SVM classifiers and CNN structures. Target objects with dense features may not be accurately identified.

Some solutions have been proposed to solve the problems of localization inaccuracy and lower efficiency. In 2014, Erhan et al. (2014) proposed a saliency-inspired neural network which is able to predict a set of class-independent bounding boxes and make predictions with a probability score for each bounding box. The model can generalize across categories at the highest level of the network. In 2015, Liang and Hu (2015) utilized a recurrent CNN model that merge the recurrent connections in each convolutional layer so that each unit is affected by its neighboring units and fully utilizes the information from the context. The network has multiple paths and fewer trainable parameters, which enhances the training process. In the same year, Ouyang et al. (2015) introduced a deformable deep CNN (DeepID-Net). This network incorporates a unique deformation constrained pooling (def-pooling) layer to enforce geometric constraints on object part deformations. In 2016, Pont-Tuset et al. (2016) proposed one bottom-up hierarchical image segmentation method named Multiscale Combinatorial Grouping (MCG) that combines multi-scale regions into high-precision target candidates by efficiently exploring assemblage Spaces. Also, some models are proposed to filter out some useless feature layers to refine the object segments in the selected datasets like DeepMask (O. Pinheiro et al., 2015; Deng et al., 2023) or DeepBox (Kuo et al., 2015).

Experiment results showed that R-CNN had the mean Average Precision(mAP) value of 58.5% in Pascal VOC 2007 dataset (Everingham et al., 2010), with a large increase from DPM (Felzenszwalb et al., 2010) framework(33.7%). In Pascal VOC 2010 dataset, the R-CNN framework achieved the mAP value of 53.7% largely higher than SegDPM (Fidler et al., 2013) in 40.4%. The R-CNN largely improved the detection performance on Pascal VOC datasets, but it cannot be trained on the GPU units for faster training.

3.1.2. SPP-net

As CNN networks mainly consist of convolutional layers and fully connected layers, they require fixed sizes in input images. However, R-CNN networks often crop or warp the input images to fit the size of fully connected layer. Although the preprocessing aligns with the network's requirement, image distortion can significantly impact image quality and recognition accuracy.

The concept of spatial pyramid matching(SPM) (Grauman and Darrell, 2005; Lazebnik et al., 2006) presents an ideal solution to the challenge of preprocessing. By combining SPM with CNN networks, the requirements for fixed input image sizes is alleviated, resulting in proposing a new network model named SPP-Net (He et al., 2015). The SPP layer separates the images into several scales from finer to coarser levels, and then aggregates local features into higher-level representations. This methods mitigates the limitations associated with fixed input sizes, which improves the flexibility and performance in object detection tasks.

The basic steps of SPP-Net structure is shown in Fig. 4, it can be seen that basic structure of SPP-Net is similar with CNN network. Unlike the sliding window pooling used in previous networks, SPP has the unique capability to produce a fixed-length output regardless of the input size. The use of multi-level spatial bins makes the object deformation more robust. The SPP-Net framework contains a series of convolutional layers, SPP layers and fully connected layers, with the connection of these elements, the SPP-Net can be more applicable to deal with images in different sizes. With the widespread use of SPP-Net structure, the feature maps inside the whole image no longer need to be calculated more than one time, and the convolutional features no longer need to be computed repeatedly.

Experiment results demonstrated that the mAP value of SPP-Net in Pascal VOC 2007 increased slightly to 59.2% than R-CNN model(58.5%). Meanwhile, the detection speed was significantly accelerated more than 20 times than R-CNN algorithm. The total GPU time in SPP-Net spent only 0.382s about 24 times than 9.03s of R-CNN, which is a significant improvement.

3.1.3. Fast R-CNN

Although SPP-Net has enhanced the efficiency and accuracy in comparison with R-CNN model, it still remains some limitations. The SPP-Net still follows the traditional pipeline of R-CNN model which includes: feature extraction, networks finetuning, classifier training and bounding box regression. However, the weights of convolutional layers cannot be updated, posing significant limitations on deep networks, and resulting in decreased accuracy. Moreover traditional multistage pipeline incurs traditional multistage pipeline. In 2015, Girshick (Girshick, 2015) proposed a faster version of R-CNN network, which is called Fast R-CNN. The network extracts images through convolutional layers first, then the extracted feature vectors along with the bounding boxes are sent into the RoI pooling layer to obtain the fixed size features. Each feature vector is sent into the input of bounding box regressor and classifier.

In the Fast R-CNN, the training process in network layers can be regarded as an one-stage end-to-end with a multi-task loss. The multi-task loss L is defined to train classification and bounding box regression jointly:

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v) \quad (1)$$

where $L_{\text{cls}}(p, u) = -\log p_u$ calculates the classification loss with ground truth u, $p_u = (p_0, \ldots, p_N)$ is computed as a discrete probability distribution over N+1 output for the fully connected layer. $L_{loc}(t^u, v)$ calculates

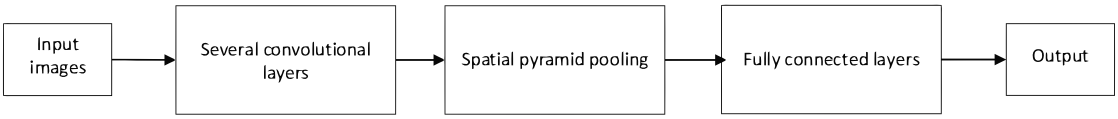


Fig. 4. Basic flow of SPP-Net.

the loss function between the ground truth bounding box regressor $v = (v_x, v_y, v_w, v_h)$, and predicted tuple $t_n = (t_x^n, t_y^n, t_w^n, t_h^n)$, where x, y, w, h represent the attributes of bounding boxes, and n represents the index of the category. The Iverson bracket indicator $[u \geq 1]$ is defined to eliminate the influence of background RoI, and the smooth L1 loss (Girshick, 2015) is utilized to make the regressor more robust to anomalies is defined as follows:

$$L_{\text{loc}}(t^u, v) = \sum_{i \in x,y,w,h} \text{smooth}_{L_1}(t_i^u - v_i) \tag{2}$$

where

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{3}$$

The utilization of the RoI pooling layer plays a crucial role in extracting a fixed-size feature map from region proposals, which effectively addressing the issue of varying sizes of RoIs. However, the detection of RoIs typically consumes large amount of time during forward propagation, by using truncated SVD (Girshick, 2015) in the fully connected layers in the network is helpful for reducing computational overhead and improving overall efficiency.

Experimental results showed that Fast R-CNN had mAP value of 70% in VOC 2007 datasets much larger than 58.5% in R-CNN method. The detection speed of Fast R-CNN(0.32s) along with the truncated SVD accelerated 213 times than R-CNN(47s) in GPU units. In MS COCO dataset (Lin et al., 2014), Fast R-CNN achieved 35.9% in AP_{50}, which is the same index with the mAP in Pascal dataset.

3.1.4. Faster R-CNN

Previous methods generate region proposals and candidate boxes using selective search or sliding windows. Although these methods can produce candidate boxes to some extent, issues such as accuracy and generation speed pose significant challenges in traditional networks. In order to solve the challenge, in 2015, Ren et al. (2015) introduced the concept of the Region Proposal Network(RPN) (Ren et al., 2015; Zhong et al., 2019), which is a fully convolutional neural network designed to generate candidate target regions, and the structure is incorporated into Faster R-CNN framework. By sharing the convolutional features with detection network, the RPN effectively minimizes the cost of region proposal, making it nearly free. It improves both the speed and accuracy of region proposal.

The RPN is implemented as a Fully Convolutional Network (FCN), enabling it to predict object bounds and scores for multiple positions simultaneously. The RPN takes an image of arbitrary size produces a set of rectangular object proposals. It operates on a specific convolutional layer, with the preceding layers shared with the object detection network. This design enables RPN to efficiently generate region proposals while leveraging the shared feature maps for subsequent object detection tasks.

The basic structure of RPN network is shown in Fig. 5. The network performs sliding operations over the convolutional feature map and fully connects to an n×n spatial window. Within each sliding window, a low-dimensional vector (512-dimensional for VGG16) is extracted and then fed into two sibling fully connected (FC) layers, namely the box classification layer (cls) and the box-regression layer (reg). This architecture is achieved using an n × n convolutional layer, followed by two sibling 1 × 1 convolutional layers. ReLU activation is applied to the output of the n × n convolutional layer to introduce nonlinearity.

Similar to Fast R-CNN, the multi-task loss is defined with classification and bounding box regressors. The parameters of minibatch size(N_cls) and number of locations(N_reg) are used to normalize the part of L_cls and L_reg in the loss function, where p_i is the probability of the ith anchor, p_i^* is the ground truth label of the anchor. t_i and t_i^* are the parameters related to the bounding box regressors, which are defined similar to p_i and p_i^*.

$$L(p_i, t_i) = \frac{1}{N_{\text{cls}}} \sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}} \sum_i [p_i^* L]_{\text{reg}}(t_i, t_i^*) \tag{4}$$

Experimental results demonstrated that Faster R-CNN improved the detection accuracy and speed than Fast R-CNN method. The detection on Faster R-CNN has the frame rate of 5fps on GPU units, and the system takes 198 ms on both proposal and detection, much faster than Fast R-CNN in 320 ms and 1.51s in SPP-Net. With the mAP value of 73.2% in VOC 2007 datasets and 70.4% in VOC 2012 datasets, much larger than 58.5% in R-CNN method. In MS COCO dataset, the AP_{50} value achieved 42.7% improved obviously than Fast R-CNN(35.9%) method.

3.1.5. Feature pyramid network(FPN)

The networks mentioned above have gradually improved the speed of feature extraction, making object detection models more robust and efficient. However, when detecting datasets containing multiscale features or small-scale objects, the results may be poorer than common objects. In order to solve the challenge, in 2017, Lin et al. (2017a) proposed the Feature Pyramid Network(FPN) to improve object detection performance with objects in scale variance.

As lower-level feature maps have higher resolution but less semantic information, while higher level feature maps have more semantic information but lower resolution. The FPN network employs a pyramid structure (Lin et al., 2017a) with button-up and top-down pathways. The button-up link is similar with convolutional structures for downsampling with a regular stride of 2, simultaneously, the extracted features in separate levels are distributed into lateral connections in support of the top-down pathway. Using the results from lateral connections, features from higher levels are upsampled and enhanced by the usage of the same spatial size with the button-up stage, which is the function of a top-down link. By integrating both pathways, semantic features can be effectively trained across multiple scales in an end-to-end manner. The structure of FPN is independent of the structure of core network (Fig. 6) so that the challenge of memory and time consumption can be perfectly solved.

Combining the FPN network with the two-stage object detection methods has become the popular topics in academic area. Typically, Faster R-CNN method utilizes the bottom-up pathway to extract features for different layers, the basic extraction procedure can be represented as $F_l = \text{Conv}_l(I)$, where F_l represents the lth feature layer and $Conv_l(I)$ means the convolution step on previous layer. The top-down pathway is generally utilized to add spatial resolution, which can be represented as $U_l = U_p(F_{l+1})$, where U_p is upsampling operation. After that, the fusion feature can be combined by lateral fusion, and the final fusion feature can be defined as $P_l = \text{Conv}_l(1 * 1) + U_l$, where P_l shows the fused layer.

After the fusion steps on whole features, the Region Proposal Network(RPN) generates proposals for different scales along with the prediction on the presence of objects and bounding box predictions. The convolutional calculation for classification and regression are defined as $cls = FC_{cls}(ROI(p_l))$ and $reg = FC_{reg}(ROI(p_l))$, the final loss

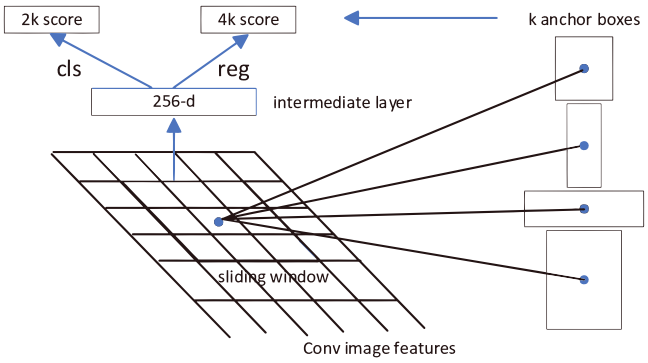


Fig. 5. Basic flow of RPN.

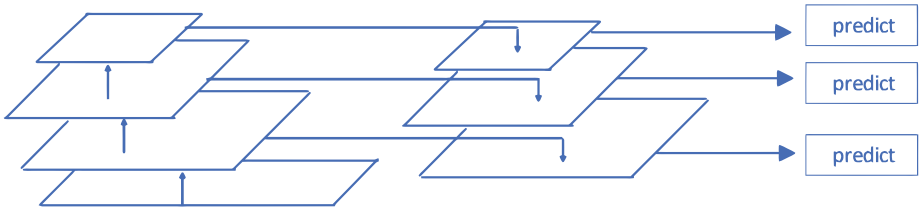


Fig. 6. Architecture of FPN.

function can be calculated as below, where L_{cls} and L_{reg} represent the classification and regression loss, cls^* and reg^* are the true labels of category and bounding box regression object.

$$L = \sum L_{\text{cls}}(\text{cls}, \text{cls}^*) + \lambda \sum L_{\text{reg}}(\text{reg}, \text{reg}^*) \tag{5}$$

With the advantages of FPN, the state-of-the-art methods primarily combine FPN with CNN-based networks, which can be widely used in multi-scale representation. In 2020, Guo et al. (2020) proposed a feature pyramid network-based structure, which utilizes residual feature augmentation to reduce the loss of higher-level information. In 2021, Zhao et al. (2021) proposed a novel graph feature pyramid network by using the superpixel hierarchy structure to find the intrinsic image structures along with feature interactions for various scales of images. In 2022, Li et al. (2022d) proposed a novel attention-based structure which takes most of the lower layers with the adjacent layer of feature fusion to filter higher layers of features. The structure is combined with FPN-based network. An improved feature pyramid network (Zhu et al., 2022) was proposed to use the similarity-based module to fuse the features for the adaptation of various sizes of instances, and the improved fusion mechanism can be used for specific tasks. In 2023, Li et al. (2023b) introduced a novel object detection algorithm, named MSFFA, which combines the attention CNN network with multi-scale fusion to extract more effective features even in complex backgrounds. Due to the pyramid structure of FPN, it has become the basic structure on handling multiscale feature fusion areas. The improved methods derived from FPN can be applied to other branches of computer vision tasks like image segmentation based on the versatility and effectiveness issues.

With the utilization of FPN network inside Faster R-CNN series model, the experimental results improved obviously in MS COCO dataset. The AP_{50} value of FPN improved significantly than Faster R-CNN network(42.7%), with the value of 59.1%. The running time of FPN was largely reduced in GPU units for only 0.165 s lower than 0.32 s in Faster R-CNN.

3.2. One-stage object detection

3.2.1. YOLO

In 2015, Redmon et al. (2016) firstly proposed the model You Only Look Once(YOLO), and opened up a new era of real-time applications. While R-CNN-based models excel in detection accuracy and computational performance, many real-time detection applications prioritize speed and timeliness. Pooling operations in certain model families can consume considerable processing time, making them less ideal for real-time analysis. With the advantage of real-time and simpler structure, the YOLO family of models can be used broadly in real-life detection scenarios (Bolya et al., 2019; Simon et al., 2019; Li et al., 2022b).

The YOLO pipeline divides each image into an $S \times S$ grid, and each grid cell detects what specific object belongs to the center of grid cells. Each grid cell predicts B bounding boxes along with their confidence scores. The bounding box consists of four properties: x, y, w, h, where (x, y) defines the center of the box, and (w, h) represents the size of the box. The confidence score contains two separate parts: Pr(Object), which defines the probability of objects belonging to the bounding box, and $IOU_{\text{pred}}^{\text{truth}}$, which shows the accuracy of the bounding box containing the object. The formula of confidence can be written as follows: $\Pr(\text{Object}) \times IOU_{\text{pred}}^{\text{truth}}$.

The network structure of YOLO is based on GoogLeNet (Szegedy et al., 2015; Huang et al., 2021), which is a general CNN structure. YOLO network contains 24 convolutional layers following 2 fully connected layers, and the inception module uses 1×1 reduction layers and 3×3 convolutional layers, which modifies the original structure of GoogLeNet. It is necessary to double the input resolution of 224×224 pixels from network so as to get fine-grained visual information (Redmon et al., 2016). The YOLO structure is not that difficult as a single convolutional network directly predict the location of bounding boxes and categories directly, which highly improves the detection speed than two-stage detectors.

However, the YOLO algorithm suffers problems of detection challenge on adjacent and small scale objects with lower accuracy of original images with multiple times of downsampling. The learning strategy can only be implemented on the basis of rough features from images. One year later, a second version of YOLOv2 (Redmon and Farhadi, 2017) was proposed to solve the problems of YOLO. YOLOv2 uses several concepts to improve the speed and accuracy, such as batch normalization, high resolution classifier, anchor boxes, dimension clusters and multi-scale training. The batch normalization in the network helps to normalize the training sets for accelerating convergence to regularize the model. The YOLOv2 network improves the resolution

of 448 × 448 pixels on fine-tuned model 10 times on ImageNet with pretrained model. Darknet-19 is the network proposed in YOLOv2, which simplifies the traditional CNN network structure by replacing the dense layers with fully convolutional layers. Furthermore, the utilization of anchor boxes and k-means clustering make better performance and detect objects more accurately. The pre-defined anchor boxes are utilized to match the original types of objects, these anchors are located inside the grid cells for predicting locations and categories on different anchors. The K-means dimension clustering is utilized in support of learning and predicting bounding boxes for finding better prior defined candidate boxes. In the same time, YOLOv2 no longer rely on fully connected layers, input images can be selected on arbitrary scales to train multi-scale features. YOLOv3 (Redmon and Farhadi, 2018) improves the function of YOLOv2, which uses multi-label classification datasets to extend the application of model in tasks with overlapping labels. YOLOv3 model redesigned the backbone network of Darknet composed of 53 convolutional layers along with the Residual connection layers. In comparison with YOLOv2 network, the network structure has been largely extended in YOLOv3, and more precise boxes can be gathered to improve the detection performance on small-scale objects. On the basis of Darknet architecture, YOLOv3 model utilizes three output layers of different sizes to detect multi-scale objects. With the appearance of YOLOv3, the detection on small-scale objects and objects with multi-scale features can be significantly improved.

Redmon's team no longer engaged in doing research in real time object detection after the advent of YOLOv3. In 2020, the improved methods based on YOLOv3 named YOLOv4 (Bochkovskiy et al., 2020; Wang et al., 2021a) was proposed, which utilized the back bone of YOLOv3, and applied the Path Aggregation Network(PANet) to realize cross-level feature fusion. PANet carry out information transfer and fusion between feature maps of different levels so that the model can detect the target by using both low-level and high-level features. The model utilize CSPDarknet53 as the backbone network, combining with PANet, YOLOv4 learns the features more precisely and achieves better performance. The CSPDarknet inside YOLOv4 model employs the Mish (Misra, 2019) activation function, a non-monotonic activation function, which provides more smooth gradients during both forward and backward propagation progress. The YOLOv4 method combines training strategies of bag-of-specials and bag-of-freebies data augmentation solution, which enlarges the reception field for better obtaining robust features. The self-adversarial training strategy inside the model implements adversarial attack for creating a deception label for maintaining the original ground truth objects. A few months after YOLOv4 model, YOLOv5 model has been released with a simplified version based on YOLOv4 model. YOLOv5 utilizes the Pytorch framework, rather than Darknet framework to develop. It is easier to use, train and deploy in most cases, therefore YOLOv5 is popular among industrial environment. In recent years, more methods are proposed and optimized on the YOLO-based family, such as YOLOX (Ge et al., 2021), YOLOv6 (Li et al., 2022a), the updated version YOLOv8 (Jocher et al., 2023) has been released recently. The YOLOX model employs an anchor-free detector and central sampling strategy to avoid the imbalance on the lack of anchor-points, which regards central 3 × 3 area as the positive example area. YOLOv6 introduces a novel self-distillation strategy for classification and regression tasks and uses queue learning methods to assign labels. The updated YOLOv8 model reduces the number of bounding boxes prediction, and accelerates the speed on Non-maximum suppression, the detection performance and speed achieves the optimum trade-off in the updated version.

Experiment results demonstrated that the YOLO series algorithms largely improved detection speed than two-stage CNN based object detection methods. Although YOLOv1 has the mAP value of 63.4% lower than Faster R-CNN's 70.4%, YOLOv1 runs at 45 frames per second(fps) much faster than 7fps in Faster R-CNN. The appearance of YOLO series methods open up a new period in real time object direction, the improved version gradually improved the detection performance, with YOLOv2 in mAP 78.6% and 19fps in Pascal datasets, 44% of AP_{50} value in COCO dataset, YOLOv3 in mAP 57.9% of AP_{50} value in COCO dataset. With the improved version of YOLO serious models, the trade-offs between speed and accuracy can be largely improved. In YOLOv5, the map has improved to 43.3% in COCO dataset along with 869 FPS, and in YOLOv8 the map value has risen to 50.2% in COCO dataset with 234.7 FPS.

As YOLO series algorithms has advantage in real-time detection, it is widely used in real-life scenarios like autonomous driving, safety monitoring, retail analysis etc. Specifically, the application on detecting extreme weather conditions seem to be an interesting topic in object detection. In 2023, Kumar and Muhammad (2023) proposed a novel advanced YOLOv8-based object detection method, which combined with transfer learning strategy for data augmentation. Through the training on images of merging different weather conditions, the results performed better than individual condition. In 2024, Gupta et al. (2024) proposed an idea of adding synthetic weather noise in case of fog, rain, and snow on utilizing illumination information. By adding noising elements into the training features of YOLOv5 model for pre-training, the trade-off between noise reduction and detection performance enhances the robustness on object detectors. The same year, Chu (2024) proposed the D-YOLO method, a novel robust object detection framework, with the utilization of attention feature fusion module to fully consider different conditions of extreme weather. Through minimizing the distance between normal conditions and hazy types, the richness of features can be selected in support of training, which enhanced the robustness of detector.

From these above papers, the weather conditions are functioned as the data augmentation methods to enhance object detection's performance. Fog augmentation, rain augmentation and snow augmentation are utilized as the analytical method of noising along with the weather denoising functions. The conditions of fog, rain and snow are commonly utilized as noising methods for the enhancement of robustness, the denoising methods can be utilized on the pre-trained weight for analysis. In majority cases, the object detection methods use clear images for training processes, so the mAP value of weather condition detection models not improved significantly. Through the usage of transfer learning and data augmentation, the issue can be majorly solved. The conditions of weather conditions are majorly relying on the outer environment, the minimum conditions are highly close to the extreme conditions of real life like sub-zero temperature of snowy weather and the optimum conditions are highly linked to the mild cases like the sunlight should avoid the sensor distortion. All the weather conditions for best operations on affecting object detection performances are listed in Table 2.

3.2.2. Single short detector(SSD)

Single Short Detector(SSD) (Liu et al., 2016) is a one-stage model for predicting multiple categories, which is another branch in parallel with the YOLO series. The SSD utilizes a predefined set of anchor boxes with varying aspect ratios and scales to discretize the bounding box output space. The network fuses predictions with multiple feature maps in different resolutions, so as to solve the challenge of detecting objects with different ratios or scales in the YOLO network.

Based on the structure of VGG16, SSD adds some convolutional feature layers to the end of network to predict detections in multiple scales. The network is trained with the weighted sum of confidence loss and localization loss, and the detection results are formed by conducting NMS. The general structure of SSD is shown as Fig. 7. The SSD model contains the backbone convolution network with VGG16 network, utilized as feature extractor with fully convolutional layers to facilitate feature mapping process. The additional feature layers are employed to capture wider features on input layers, which enables detecting objects with different scales. By utilizing the extracted features from both backbone network and additional layers, the SSD model has

Table 2
Weather conditions affecting real-time object detection.

Weather type	Minimum conditions	Optimum conditions
Sun	Direct bright light or extreme low light may cause great challenge.	Bright and no direct light, such as cloudy days or dusk times.
Rain	In heavy rains or storms, large raindrops may disturb the visibility.	Conditions on slight rain or after rainfall may have stable visual condition.
Fog	Heavy fog condition may cause huge challenge on detection performance with low visibility.	The visibility may be better in the conditions of light fog or post-fog.
Storm	Extremely light level conditions may lead to obstruction on visual signals.	The utilization of radar or sensors after heavy storms may have better resolution.

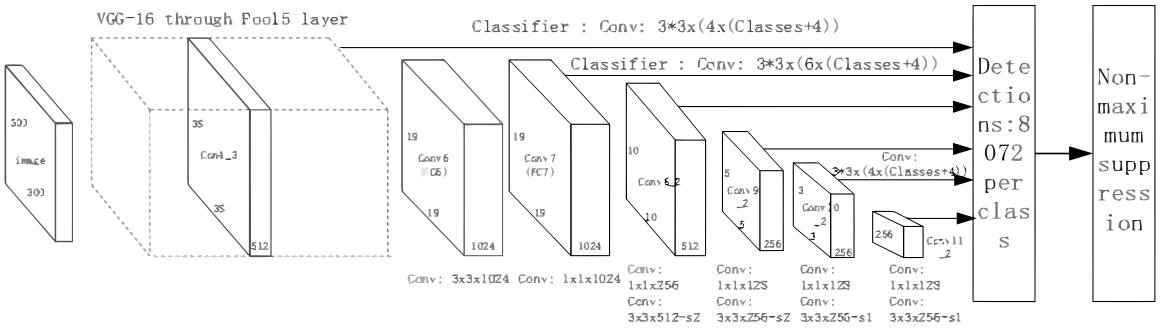


Fig. 7. Network of SSD.

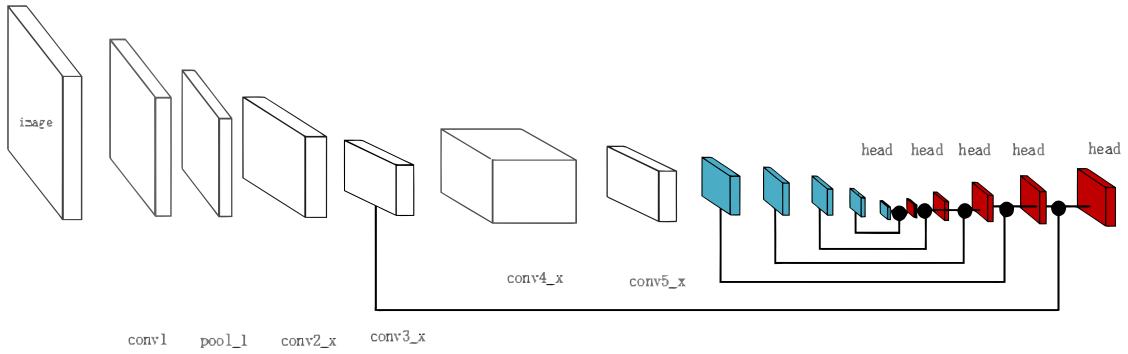


Fig. 8. Network of DSSD.

the advantage on detecting objects with multiple scales better than the original YOLOv1 version.

Deconvolutional Single Shot Detector(DSSD) (Fu et al., 2017) is an improved version of SSD network, which utilizes the Residual-101 network instead of VGG to improve accuracy. Similar with the SSD network, the DSSD model contains backbone network and additional convolutional layers to extract multi-scale features on input images. The deconvolution layer aims at increasing the resolution of feature maps, through the upsampling process on each additional convolutional layer to recover the loss of detailed information caused by repeated convolution and pooling operations. After the sampling processes, the high-level feature information are fused with corresponding lower-level features through pixel-level fusion methods on different modules. In prediction layers, a residual block is added to learn the residual or the difference between the input and the output of a layer. The training process utilizes the pre-trained ResNet-101-based model, and then uses the input sizes of 321 × 321 or 513 × 513 to train the original SSD model on the detection dataset. Finally, the authors train the deconvolution module while keeping all the weights of the SSD module frozen. The basic structure of deconvolution and DSSD are shown as Fig. 8.

Experiment results showed that SSD achieved a competitive result on mAP value and detection speed. SSD had mAP of 76.8% in VOC 2007 dataset and 74.9% in VOC 2012. The DSSD method obviously improved the mAP of 81.5% in VOC 2007 and 80.7 in VOC 2012. The performance improved a lot than those Fast R-CNN series models, with SSD achieved 22 FPS higher than the original YOLO method and the DSSD model.

3.2.3. RetinaNet

RetinaNet (Lin et al., 2017b) is another one-stage detector including focal loss and FPN (Lin et al., 2017a). As one stage detectors have lower accuracy due to the class imbalance, the FPN obtains feature maps at different scales by establishing lateral connections and top-down connections between different levels. Furthermore, the focal loss function uses focusing parameter to reduce the weight of easily classified samples effectively, so that the network is more able to focus on samples hard to classify. With the use of focal loss, the RetinaNet network can balance losses when handling a large number of easy and difficult to classify samples simultaneously. In this case, the problem of solving imbalanced data and obviously improves the performance of detection without losing the speed of the one-stage methods. The RetinaNet model is the typical FPN-based multi-scale detector, the major components inside the model contains both bottom-up pathway and top-down pathway along with the lateral connections to merge the features in separate layers. After the fusion process on different scales, the sampled features are transferred into classification network and regression network separately for predicting the category

Table 3
Major contributions on CNN-based models.

Category	Method	Contributions and innovations
two-stage detectors	Fast R-CNN (Girshick, 2015)	The model utilizes the Region of Interest(ROI) pooling layers to extract feature vectors in a fixed size.
	Faster R-CNN (Ren et al., 2015)	The model introduces the region proposal network(RPN) to generate region proposals in different scales and ratios.
	OHEM (Shrivastava et al., 2016)	The paper combines OHEM with Faster R-CNN detector for improving the performance on class imbalance training issue.
	ION (Bell et al., 2016)	By integrating the contextual information through spatial recurrent neural networks to capture long-range dependencies.
	HyperNet (Kong et al., 2016)	The model utilizes hyper features to extract multi-level features for better understanding multi-scale semantic information.
	CoupleNet (Zhu et al., 2017)	The CoupleNet better combines global and local context features through coupling modules for comprehensive understanding.
	Mask R-CNN (He et al., 2017)	Mask R-CNN uses ROIAlign for better alignment and integrates FPN network to construct pyramid for multi-scale features.
	Cascade R-CNN (Cai and Vasconcelos, 2018)	Cascade R-CNN employs the cascade training process to solve the degradation of performance with increased IoU threshold.
	Grid R-CNN (Lu et al., 2019)	Grid R-CNN divides images into grid of cells for adjusting the grid sizes within scales of objects inside images.
	Augfpn (Guo et al., 2020)	The proposed method Augfpn employs consistent supervision to narrow the semantic gap between different scales before fusion.
one-stage detectors	RetinaNet (Lin et al., 2017b)	Focal loss inside the paper utilizes the adaptive weighting scheme to modulate the focal loss in a dynamic style.
	YOLOv3 (Redmon and Farhadi, 2018)	The model utilizes anchor boxes to predict the location and size of objects. Results on small-scale objects are greatly improved.
	RefineDet (Zhang et al., 2018)	The refinement module inside the paper leverages feature fusion and contextual information to improve prediction performance.
	Cornernet (Law and Deng, 2018)	The CenterNet method utilizes corner pooling mechanism to fully capture the corner feature of the target from the feature map.
	CenterNet (Zhou et al., 2019a)	The CenterNet method utilizes key-point and center-point based detection paradigm to consider objects into single points.
	ExtremeNet (Zhou et al., 2019b)	The utilization of extreme points and center points can directly identifying key points and bounding boxes in object detection.
	YOLOX (Ge et al., 2021)	YOLOX introduces a novel label allocation strategy on the basis of Optimal Transport Assignment for improving allocation precision.
	YOLOv6 (Li et al., 2022a)	YOLOv6 redesigns the framework in various sizes to achieve the trade-offs between speed and accuracy in industry places.
	YOLOv7 (Wang et al., 2023)	The YOLOv7 method in this paper introduces adaptive anchoring strategy for better adaptation on different scales and portions.
	YOLOv8 (Jocher et al., 2023)	The YOLOv8 method introduces the Anchor-Free detectors to simplify the training speed and the model is more flexible.

of objects. With the utilization of FPN network structure, the detection performance on small-scale objects has been significantly improved while the detection speed has not been affected greatly.

Experiments showed that RetinaNet got the AP value of 39.1% in comparison with the DSSD model of 33.2% in COCO dataset and achieved significant improvement in small scale objects with 21.8% much larger than DSSD(13%). The performance and speed trade-off has been improved among DSSD method, with 39.1% AP and 5 fps in COCO dataset.

Above all are the major categories of CNN-based object detection methods, however, more detailed variations among these methods are not illustrated clearly among these branches. The detailed experimental results are listed in Section 5, and the contribution and innovation points among these variations are demonstrated in Table 3.

4. Transformer-based models

Although transformer models have been used widely in NLP models like BERT (Devlin et al., 2018), object detection still requires prior knowledge and predefined bonding boxes. The common detectors still waste too much time on postprocessing step, which utilize non-maximum suppression to filter out redundant bounding boxes, and the detection process is still completed. It was not until 2020 when Carion et al. (2020) proposed the DEtection TRansformer(DETR) model by defining object detection problems into set-based tasks. The appearance of DETR is a revolutionary stage in computer vision, which simplifies the object detection into an end-to-end framework by using the encoder–decoder module of transformer. The transformer-based models can be divided into DETR based model and ViT based model, the former removes the NMS module which is the hand-crafted process and the latter divides the whole images into patches, where each path contains a piece of the image area.

4.1. End-to-end object detection

4.1.1. DETR

The CNN-based detectors still waste too much time on the post-processing step, which utilizes non-maximum suppression to filter out

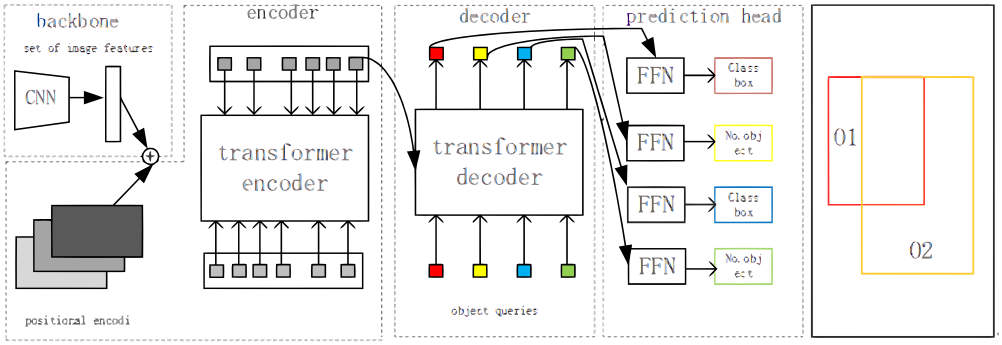


Fig. 9. workflow of DETR.

redundant bounding boxes, and the detection process is still complicated. It was not until 2020 that the DETR model converted object detection problems into set-based tasks. The appearance of DETR is a revolutionary stage in computer vision, which simplifies object detection into an end-to-end framework by using the encoder–decoder module of transformers.

As modern detectors utilize proposals or anchors to do surrogate tasks, the results are heavily influenced by postprocessing progress. The method transforms the task into a direct set prediction and use the encoder–decoder structure in transformer, and also models the relationship. By combining the bipartite loss among predictions and ground-truth boxes and parallel decoding together, the hand-crafted features are removed in common detectors. The DETR structure contains four major parts: CNN backbone, transformer encoder, transformer decoder and final prediction.

The backbone structure of the model uses conventional CNN structure to extract 2D features from input images, and the model flattens it and matches with positional encoding. The transformer encoder use the sequence features as the input, which learns global characters of images and the decoder use the output of encoder along with object queries for guiding model's attention to different objects. Finally, the detection results of decoders are transferred into the feed forward network(FFN) to predict specific objects. The flow of DETR is shown as Fig. 9.

The appearance of DETR significantly simplifies the overflow of object detection, and the model is easy to extend to various areas of computer vision tasks. The structure of DETR is easy and flexible, which achieve better results on large scale objects. However, the model has lower accuracy on small scale objects and the speed of convergence is slow, which needs over 500 epochs. Further researches are based on DETR architecture, and gradually improved the performance of model.

Experiment results represented that with the increment of parameters inside DETR model, the mAP value achieved 35.3% at 50 training epochs and 43.3% after 500 epochs. The improvement of AP_L in large-scale object was significant, with 61.8% much larger than Faster R-CNN method. Results in small-scale objects still need to improve in further progress, and the amount of training epochs need to be optimized in future researches. The GFLOPS index of DETR is 187, much lower than Faster R-CNN, which largely saves the computational costs.

4.1.2. Deformable DETR

One year later, an improved version of DETR has developed to overcome the issues of low accuracy on small objects and requires too many learning epochs. In 2020, Zhu et al. (2020) proposed a novel Deformable DETR network which utilized the concept of deformable convolutional network(DCN) (Dai et al., 2017; Zhu et al., 2019) to improve the architecture of DETR. The new model use the characteristics of deformable to solve the issue of how to find the location of objects when the they are obscured or deformed by adapting the location of sampling.

The deformable attention model only uses a small number of key points surrounding each reference point. Problems of convergence and feature spatial resolution can be partially overcome by using fixed number of keys for query, and the deformable attention feature can be calculated as below:

$$\text{Deformable}(z_q, p_q, x) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \tag{6}$$

In a feature map of $x \in \mathbb{R}^{C \times W \times H}$ parameter q represents the index of the query of content feature z_q and 2-d reference point p_q. In the formula, m represents the attention head, and Δp_{mqk}, $A_m qk$ shows the sampling offset and attention weight of the $k^t h$ sampling point in the $m^t h$ attention head.

The multi-scale feature pyramid is also employed in the Deformable DETR network. Similar to the FPN structure in two-stage detectors, the model fuses the multi-scale feature fusion to improve the ability to perceive different sizes. By combining the end-to-end structure with multi-scale feature fusion, the model has the capability to learn abundant feature expressions of various levels.

With the improvement of the DETR model, the Deformable DETR not only improves the performance on efficiency and accuracy but also saves a huge amount of time on training epochs. Experiment results demonstrated that with the application of deformable layers, the detection accuracy can be improved for learning features from neighboring features. The performance on small scale objects improved greatly in comparison with original DETR method, with 25.1% on AP_S value for small-scale objects larger than DETR(22.5%). With the implementation of multi-scale feature maps, the detection accuracy improved to 46.9% and the detection speed was largely accelerated with 1.6 times faster than DETR method. Simultaneously, the GFLOP of Deformable DETR has also been degraded to 173.

4.1.3. Efficient DETR

Both the DETR and Deformable DETR models require a cascading structure of 6 decoder layers to achieve ideal performance, as random initialized object query may not provide a good state. The proposal of Efficient DETR (Yao et al., 2021) simplifies the structure of the 6-encoder and 6-decoder in former methods. As the encoder layers have less influence for detecting results, cutting down the number of encoders is acceptable to simplify the structure of the model. By combining dense detection and sparse detection together, the Efficient DETR utilizes dense detection before initializing the object container, which narrows the gap between the 1-decoder and 6-decoder structure.

The model uses 3 encoder layers and only 1 decoder layer, without using a cascade structure for the decoder. The dense part utilizes RPN to select the top-k set of proposals from dense features, and the initialization of object queries are taken from the backbone, encoder, and detection head. For the sparse part, the reference points, object containers along object queries from the output of the dense part are sent into the decoder to interact with encoder features. The final results are predicted by the information from the detection head by the extracted object container.

Table 4
Major contributions on DETR-based models.

Category	Method	Contributions and innovations
DETR series	Deformable DETR (Zhu et al., 2020)	The model employs the deformable attention module to focus on small partials of key points around the sampled features.
	Pix2Seq (Chen et al., 2021)	The model transforms object detection into sequence prediction problem, which simplifies the form of object detection.
	PnP-DETR (Wang et al., 2021b)	The method introduces the poll and pool sampling model for better combining with other feature extractors.
	Efficient DETR (Yao et al., 2021)	Efficient DETR employs both dense and sparse module with same detection head. The detection speed has been largely improved.
	YOLOS-DETR (Fang et al., 2021)	The method explores the potential on the transferability of transformer structure into more usage into the general model structure.
	Sparse DETR (Roh et al., 2021)	The Sparse DETR introduces the sparse sampling strategy to process only the most relevant predictions during training.
	Anchor DETR (Wang et al., 2022b)	The method utilizes anchor points as object queries for predicting multiple objects in one region, which can locate the object more precisely.
	DINO (Caron et al., 2021)	The model utilize knowledge distillation to train teacher and student models with the same network and weights.
	Focus-DETR (Zheng et al., 2023)	The method employs dual attention module to capture more detailed foreground information to enhance the fine-grained tokens.
	RT-DETR (Lv et al., 2023)	This paper redesigns an efficient hybrid encoder to handle multi-scale features through decoupling cross-scale interaction and fusion.

Comparing with previous methods, the Efficient DETR needs fewer training epochs to achieve similar results. Utilization of sparse attention mechanisms largely reduced the computational costs for attention mechanism. The combination of dense and sparse parts enables the model performs better on dense scene, and easy to adapt to different scenarios. Experimental results showed that training epochs can be reduced to 36 epochs much smaller than original DETR method(500 epochs), and the performance not change too much with the Deformable DETR method. Similarly, the computational cost in GFLOP has also been simplified to 159 in common ResNet50 network.

4.1.4. Sparse DETR

Sparse DETR (Roh et al., 2021) is another optimized method to improve the performance of the DETR model, which aims to solve the problem of overcalculation of encoders. Although Deformable DETR utilizes deformable attention to relieve the computation complexity of global attention of DETR, the token queries of the encoder increase a lot with the usage of a multi-scale mechanism. The complexity of encoder attention still remains high. It is common that partially updating tokens in the encoder does not influence the detector's performance too much, so the authors proposed the idea of encoder token sparsity to calculate the self-attention of top-k tokens for each encoder.

In order to achieve token sparsification, the Objectness Score and Decoder cross-attention Map(DAM) are proposed to find saliency regions. The former method just adds a detection header behind the feature map and then monitors it with Hungarian loss to select the higher scores for subsequent attention computation. However, the operation is independent of the decoder and does not take the decoder into account. The authors utilize DAM with cross attention of the decoder as an evaluation metric.

The DAM method aims to reserve the tokens with high response values to object queries, which are of great importance to detection results. The result of DAM for dense attention can be calculated by just directly adding each layer of cross attention of the decoder together, and the result of deformable attention can be calculated by finding the corresponding offset attention of each object. By binarizing the DAM value and using the BCE loss function to supervise the scoring network, the DAM value can be predicted. The experiment results indicated that the value of AP_S was 32.0% obviously improved with previous methods for 22.5% in DETR and 29.1% in Deformable DETR methods. The AP value was also largely increased to 49.3%, more than 5% larger than DETR method.

$$L_{\text{dam}} = -\frac{1}{N}\sum_{i=1}^{N} \text{BCE}(g(x_{\text{feat}})_i, \text{DAM}_i^{\text{bin}}) \tag{7}$$

Above all are the major contributions on DETR series object detection methods, the detailed variations on DETR-based object detection methods are generally listed in Table 4 and the experimental results are presented in Section 5. The GFLOP value remains at the same level with previous methods (with the value of 136).

4.2. Vision transformer

Vision Transformer(ViT) (Dosovitskiy et al., 2020) is another branch of application of transformer structure into image classification, which segments images into fixed-size blocks and flattens these blocks as sequences for the input of the transformer. The idea is of great reference value to the object detection area, and then multiple branches are evolved. The ViT series models are commonly used in the backbone of object detection methods, which is another new breakthrough among object detection era. Li et al. (2022c) explored the possibility of utilizing ViT structure as the backbone structure of object detection tasks, without using the FPN network. The branches among Vision Transformer brings more possibility for DETR detectors.

4.2.1. Vision transformer(ViT)

In 2020, Dosovitskiy et al. (2020) first proposed the ViT model, which enables transformer model processing image data without several convolutional stages. The model uses transformer encoder in support of image classification, and the major layer includes patch embedding, transformer encoder, and MLP head.

It converts the original 2-D image into a series of 1-D patch embeddings. The input images can be defined as $x \in \mathbb{R}^{H\times W\times C}$, where H, W represent the height and width of images, C defines the number of

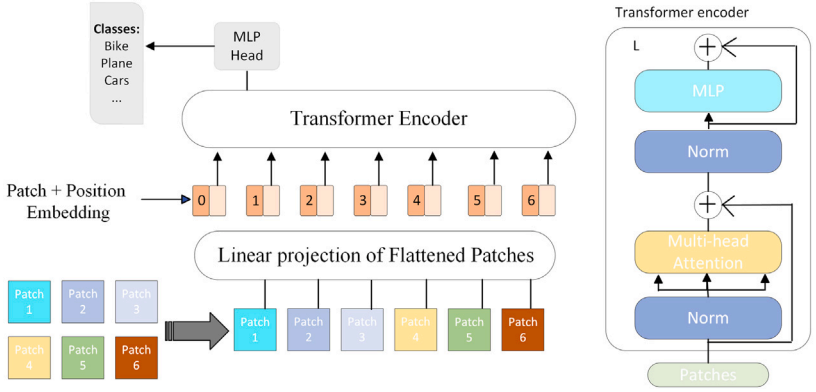


Fig. 10. General structure of ViT.

channels, P is the resolution of each patch, each image is divided into $N = HW/P^2$, also the length of sequences. Then add the special tokens to each patch, so as to fit for the structure of the transformer and avoid the preference for a single token. The position embeddings are added to the patch embeddings to retain the position information of the original images.

The results of patching and positioning embedding are used as the input of the transformer encoder. Similar to the common transformer model, the encoder layer contains several layers of multi-head attention and MLP blocks. The layer norm is introduced to dispose the of tokens. The output of the encoder remains the same size as the input and is sent to MLP Head for classification. The general structure of ViT is shown in Fig. 10.

4.2.2. Data-efficient image transformers(DeiT)

Although the ViT model makes a great contribution on converting transformer structure into the image processing area, there are still some limitations need to be improved. When dealing with high-resolution images, a growing number of plots results in the increasing length of sequence. The limitation of global self-attention mechanism may make the model hard to detect the local information of image. In 2021, Touvron et al. (2021a) proposed a novel vision transformer method named Data-efficient image Transformers(DeiT), which introduces the concept of knowledge distillation and data augmentation to improve the generalization ability and performance of model.

Similar to the ViT model, the DeiT model uses transformer structure without convolution layers with flattened patches and positional encoding as the input of the encoder. Without using MLP head attention, a simple linear classifier is employed. A fully connected layer is added to the top of the model in support of the classification task, and the header maps the output of the last encoder layer for categories in support of classification. The model uses knowledge distillation to transfer the knowledge of better results from teacher models to student models, which helps student models gather more information from teacher models to improve performance.

In 2015, Hinton et al. (2015) first proposed the distillation technique for training separate models with the same datasets, in order to make models more efficient for tasks with limited resources. The DeiT model covers two types of distillation methods: soft distillation and hard distillation. Soft distillation models use predicted labels of teacher models as targets to train student models, which aims to guide student models to capture more knowledge.

$$L_{\text{global}} = (1-\lambda)L_{CE}(\varphi(Z_s), y) + \lambda\tau^2 \text{KL}(\varphi(Z_s/\tau), \varphi(Z_t/\tau)) \tag{8}$$

where Z_s and Z_t represent the logits of the student model and teacher model, τ is the temperature of distillation, λ is the balance factor of cross-entropy loss ($L_C E$) and Kullback–Leibler Divergence (KL), and Φ is the softmax function.

The hard distillation models directly pass the predictions from teacher models to student models, and student models try to replicate the decisions of teacher models.

$$L_{\text{global}}^{\text{hardDistill}} = \frac{1}{2}L_{CE}(\varphi(Z_s), y) + \frac{1}{2}L_{CE}(\varphi(Z_s), y_t) \tag{9}$$

where $y_t = argmax_c Z_t(c)$ is the hard decision of the teacher, which plays the similar rule of ground truth in hard distillation.

The distillation token is similar to the class token, teacher model inserts the mechanism into student models to augment the quality of data. With the help of knowledge distillation and data augmentation, the performance improves a great deal under resource-limited conditions.

4.2.3. Class-attention in image transformers(CaiT)

The DeiT method successfully improves the performance of ViT method by using the knowledge of the distillation technique, but the study of optimizing network architecture has still rarely been notified. Meanwhile, how well a network is optimized is strongly related to its architecture. Based on the framework of ViT. In 2021, Touvron et al. (2021b) proposed a novel Class-Attention in Image Transformers(CaiT) structure to normalize the architecture and initialize the training weight better. The CaiT utilizes LayerScale to make deep vision transformers easy to converge and improve detection. Class-attention layers are employed to dispose of class tokens more efficiently.

LayerScale mechanism remains layer normalization in ViT and DeiT models and multiplies the output of self-attention of a feed-forward network(FFN) by a diagonal matrix. It multiplies different channels with separate parameter λ.

$$x'_l = x_l + \text{diag}(\lambda_{l,1}, \ldots, \lambda_{l,d}) \times \text{SA}(\ln(x_l)) \tag{10}$$

$$x_{l+1} = x'_l + \text{diag}(\lambda'_{l,1}, \ldots, \lambda'_{l,d}) \times \text{SA}(\ln(x'_l)) \tag{11}$$

Where $\lambda_{l,1}, \ldots, \lambda_{l,d}$, $\lambda'_{l,1}, \ldots, \lambda'_{l,d}$ are learning parameters, with the network goes deeper after 24 layers, the learnable weights are initialized as $10^{(-6)}$, and set as $10^{(-5)}$ between 18 and 24 layers. It aims at making each block closer to Identity mapping at the beginning and learning the block's own function gradually.

The class attention layers are inserted into the transformer structure to replace the last partial of networks. As the ViT structure guides attention process to find the attention map and input tokens into classifiers to do classification. It is hard to complete both tasks successfully in class embedding, so the CaiT model turns the last 2 layers into class-attention which only contains information from patch embedding to class embedding. It highly improves the efficiency of disposing of class tokens.

With the modification of network framework, the model achieves better results on transfer learning. It highly simplifies time complexity of former ViT based models.

4.2.4. Convolutional vision transformer (CvT)

Previous ViT-based models perform successfully in processing images, however, the self-attention mechanism may find it difficult to capture the local features of an image. It is a good idea to incorporate the convolution stage into vision transformer. In 2021, Wu et al. (2021) proposed a novel CvT structure, which combines convolutional layers with a self-attention mechanism so that CvT framework is able to capture both local features from convolutional stages and global contexture features in attention layers.

CvT introduces a set of convolutional layers in the initial input part of the ViT model. These layers are responsible for the initial feature extraction of the image to capture the local features of the image. After the convolution operation, CvT divides the image into multiple patches, each patch generates a local feature map. The global features are learned by transformer layer similar to ViT model. By combining local convolutional and global transformer operations, CvT can extract features at different levels and scales. With the inherent incorporation of local contextual patterns brought about by convolutions, CvT eliminates the need for positional embedding. This characteristic potentially positions it as advantageous for accommodating diverse vision tasks that involve varying input resolutions.

5. Datasets and evaluation

Building large datasets is of great importance to compare the performance between algorithms and find solutions to improve. Many well-known datasets have been released in recent decade for different branches, such as Pascal VOC (Everingham et al., 2010, 2015), MS COCO (Lin et al., 2014), ImageNet (Russakovsky et al., 2015) in general object detection, KITTI (Geiger et al., 2012), Cityscapes (Cordts et al., 2016) in autonomous driving, PartNet (Mo et al., 2019) for 3D detection, etc. This section mainly describes the publicly open datasets in object detection and compares the performance of different methods described in previous section.

General metrics: For general object detection methods, commonly the average precision is employed to evaluate the performance of detection and classification for object detection methods in the early stages. Normally the average precision is composed of precision and recall indexes, where precision represents the relevance of data points and recall shows the accuracy of the model related to the given data. The intersection over union(IoU) value is calculated by dividing the overlap value of prediction between bounding boxes and ground truth with the union value. Normally, in the Pascal datasets, the threshold is commonly set as 0.5, if the value is greater than the threshold, the object can be defined as "successfully detected". The "mean average precision" (mAP) value averages over all classifications is commonly used as the metrics of the detector's performance. For the MS-COCO dataset, AP is commonly identified to calculate the relative value on different IOU threshold values. Normally, mAP values are generally used for 0.5 and 0.75 and the threshold is set between [0.5,0.95], the indexes of AP, AP_{50}, AP_{75}. As MS-COCO datasets contain more detailed objects on different scales, the index of AP_S shows the AP value for small objects in the area smaller than 32^2, AP_M represents AP value for objects of area between 32^2 and 96^2, and AP_L defines the AP value of large objects with area larger than 96^2.

Pascal VOC: Pascal Visual Object Classes(VOC) is one of the classic general object detection datasets, which has been developed in the early period of computer vision from 2005 to 2012. Pascal VOC 2007 and 2012 are most commonly used in object detection, where the former contains 20 object classes and 9k images, and the latter contains 20 categories of object and 11k images along with 27+ annotated objects, an upgraded version of Pascal VOC 2007. Pascal VOC dataset has become one of the important benchmarks in computer vision areas, larger and more complex datasets have also becoming the mainstream as time goes on. Detection results on Pascal VOC datasets mainly rely on mean average precision(mAP) value, and the results on test sets in above section is shown in Table 5. In the early period of object detection, Pascal VOC has been the baseline of traditional CNN-based object detection methods. It can be seen clearly that two-stage detectors have higher mAP values than one-stage detectors.

MS-COCO: MS COCO is the challenging large-scale dataset in computer vision area, which has been widely used in detection and segmentation tasks. It originated from the Microsoft COCO dataset funded by Microsoft in 2014, which provides larger scale of data and more diverse scenarios. It contains over 330k images from 80 categories, and each image is annotated with each object's bounding boxes, labels, and key information. Comparing with Pascal VOC dataset, the images in the MS-COCO dataset are more challenging, including occluded objects, small objects, dense groups of objects, etc. This makes the algorithm need to have better generalization ability. The MS-COCO dataset has gradually become the accepted standard in object detection. As COCO dataset contains more detailed information on scales of objects, it is important to extend more indexes on evaluating the performance of object detection. AP_S, AP_M, AP_L represent the average precision values based on scales of objects, while AP, AP_{50}, AP_{75} show the indexes on separate threshold values. The detection results are clearly shown in Table 6. It is clearly seen that with the introduction of transformer structure in object detection, the DETR series object detection methods improve the performance of COCO dataset. The hyperparameters and training epoches booming largely with the improvement of computer devices.

ImageNet: ImageNet is another challenging large-scale benchmark dataset to promote the development of object detection algorithms. The ImageNet was first used for ImageNet Large Scale Visual Recognition Challenge(ILSVRC), which aims at evaluating the performance of computer vision models on object classification and location tasks. ILSVRC2014 contains 21k categories, each of which includes over hundreds of thousand images. These images cover a variety of scenes and angles in the real world, which makes the model need to have good generalization ability.

KITTI: KITTI is the widely used comprehensive dataset in object detection and tracking, especially focus on urban driving scenes. It contains data from images, laser point cloud, and Lidar and image projection, which mainly annotate the bounding boxes between cars, pedestrians, bikes, etc. Although KITTI is not as large enough as MS-COCO or ImageNet, just about 7k images, it is of great importance for the research of autonomous driving and computer vision, as the data related to urban driving can be used to verify and evaluate the performance of methods in real-time places.

Cityscapes: Cityscapes is another open and challenging dataset, which is popular among driving scenes of different cities. The dataset contains approximately 5k high-resolution images from different viewpoints, times, and weather conditions. It covers a variety of different object classes and road structures, so they can be widely used in semantic segmentation, instance segmentation, road detection. Annotations in images contain pixel-level labels, which are easy to adapt to multiple scenarios and understanding tasks.

PartNet: PartNet is a steady, large-scale dataset popular among three-dimensional objects, which has been introduced to motivate the study of the decomposition on three-dimensional objects. The dataset contains approximately 26k three-dimensional models covering 24 categories including furniture, vehicles, animals, etc. When using the PartNet dataset for target detection, the main focus is on detecting 3D objects partially, as detecting different parts of the object, rather than just detecting the whole object.

6. Conclusion and future expections

Object detection is one of the important branches in the computer vision area, and deep learning-based object detection methods have made prominent improvements in recent years. The paper shows the general overview of deep learning-based methods on different categories, dominant datasets, and detailed descriptions of the mainstream

Table 5
Detection results on Pascal VOC dataset.

Method	Backbone	Data	mAP (%)	
			VOC2007	VOC2012
two-stage detectors				
R-CNN (Girshick et al., 2014)	VGG-16	trainval	58.5	53.3
SPP-NET (He et al., 2015)	VGG-16	train	59.2	–
Fast R-CNN (Girshick, 2015)	VGG-16	trainval	70.0	68.4
Faster R-CNN (Ren et al., 2015)	VGG-16	trainval	73.2	70.4
OHEM (Shrivastava et al., 2016)	VGG-16	trainval	78.9	76.3
MR-CNN (Gidaris and Komodakis, 2015)	VGG-16	trainval	78.2	73.9
R-FCN (Dai et al., 2016)	ResNet-101	train	80.5	77.6
CoupleNet (Zhu et al., 2017)	ResNet-101	trainval	82.7	80.4
HyperNet (Kong et al., 2016)	VGG-16	trainval	76.3	71.4
ION (Bell et al., 2016)	VGG-16	trainval	79.2	76.4
LocNet (Gidaris and Komodakis, 2016)	VGG-16	traintest	78.4	74.8
DFPR (Kong et al., 2018)	ResNet-101	trainval	82.4	81.1
R-FCN++ (Li et al., 2018a)	ResNet-101	trainval	82.1	80.6
one-stage detectors				
YOLOv1 (Redmon et al., 2016)	VGG-16	trainval	63.4	57.9
SSD (Liu et al., 2016)	VGG-16	trainval	76.8	74.9
YOLOv2 (Redmon and Farhadi, 2017)	Darknet-19	trainval	78.6	73.4
DSSD (Fu et al., 2017)	ResNet-101	trainval	81.5	80.0
RON (Kong et al., 2017)	VGG-16	trainval	81.3	80.7
STDN (Zhou et al., 2018)	DenseNet	trainval	80.9	–

Table 6
The detection results of COCO datasets.

Method	Backbone	Datasets	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
two-stage detectors								
Fast R-CNN (Girshick, 2015)	VGG-16	train	19.7	35.9	–	–	–	–
Faster R-CNN (Ren et al., 2015)	VGG-16	trainval	21.9	42.7	–	–	–	–
OHEM (Shrivastava et al., 2016)	VGG-16	trainval	22.6	42.5	22.2	5	23.7	37.9
R-FCN (Dai et al., 2016)	ResNet-101	trainval	29.9	51.9	–	10.8	32.8	45
ION (Bell et al., 2016)	VGG-16	train	23.6	43.2	23.6	6.4	24.1	38.3
Faster R-CNN+++ (He et al., 2016)	ResNet-101	trainval	34.9	55.7	37.4	15.6	38.7	50.9
DCN (Dai et al., 2017)	Aligned-Inception-ResNet	trainval	37.5	58	40.7	19.4	40.1	52.5
DFPR (Kong et al., 2018)	ResNet-101	trainval	34.6	54.3	37.3	14.7	38.1	51.9
FPN (Lin et al., 2017a)	ResNet-101	trainval35k	36.2	59.1	39	18.2	39	48.2
AugFPN (Guo et al., 2020)	ResNet-101	train	38.7	61.2	41.9	24.1	42.5	49.5
Mask R-CNN (He et al., 2017)	ResNeXt-101	trainval35k	39.8	62.3	43.4	22.1	43.2	51.2
Grid R-CNN (Lu et al., 2019)	ResNeXt-101	trainval	43.2	63	46.6	25.1	46.5	55.2
Cascade R-CNN (Cai and Vasconcelos, 2018)	ResNet-101	trainval35k	42.8	62.1	46.3	23.7	45.5	55.2
PANet (Liu et al., 2018)	ResNeXt-101	trainval	47.4	67.2	51.8	30.1	51.7	60
one-stage detectors								
YOLOv2 (Redmon and Farhadi, 2017)	DarkNet-19	trainval35k	21.6	44	19.2	5	22.4	35.5
SSD (Liu et al., 2016)	ResNet-101	trainval35k	31.2	50.4	33.3	10.2	34.5	49.8
YOLOv3 (Redmon and Farhadi, 2018)	DarkNet-53	trainval35k	33	57.9	34.4	18.3	35.4	41.9
DSSD (Fu et al., 2017)	ResNet-101	trainval35k	33.2	53.3	35.2	13	35.4	51.1
RetinaNet (Lin et al., 2017b)	ResNet-101	trainval35k	39.1	59.1	42.3	21.8	42.7	50.2
RefineDet (Zhang et al., 2018)	ResNet-101	trainval35k	41.8	62.9	45.7	25.6	45.1	54.1
CornerNet (Law and Deng, 2018)	Hourglass-104	trainval35k	42.1	57.8	45.3	20.8	44.8	56.7
ExtremeNet (Zhou et al., 2019b)	DEXTR (Maninis et al., 2018)	trainval	43.7	60.5	47	24.1	46.9	57.6
RON (Kong et al., 2017)	VGG-16	trainval	25	46.5	25.4	–	–	–
DETR detectors								
DETR-DC5 (Carion et al., 2020)	ResNet-50	trainval123k	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101 (Carion et al., 2020)	ResNet-101	trainval123k	43.5	63.8	46.4	21.9	48	61.8
Pix2seq-R50(Chen et al., 2021)	ResNet-50	trainval123k	43	61	45.6	25.1	46.9	59.4
Pix2seq-R101 (Chen et al., 2021)	ResNet-101	trainval123k	44.5	62.8	47.5	26	48.2	60.3
Deformable DETR (Zhu et al., 2020)	ResNet-50	trainval123k	46.9	66.4	50.8	27.7	49.7	59.9
Deformable DETR (Zhu et al., 2020)	ResNet-101	trainval123k	48.7	68.1	52.9	29.1	51.5	62
ACT+MTKD (Zheng et al., 2020)	ResNet-50	trainval123k	43.1	–	–	22.2	47.1	61.4
PnP-DETR (Wang et al., 2021b)	ResNet-50	trainval123k	43.1	63.4	45.3	22.7	46.5	61.1
Efficient DETR-R50(Yao et al., 2021)	ResNet-50	trainval123k	45.1	63.1	49.1	28.3	48.4	59
Efficient DETR-R101 (Yao et al., 2021)	ResNet-101	trainval123k	45.7	64.1	49.5	28.2	49.1	60.2
Sparse DETR (Roh et al., 2021)	Swin-B	trainval123k	49.3	69.5	53.3	32	52.7	64.9
DINO (Caron et al., 2021)	ViT-S	trainval123k	49.4	66.9	53.8	32.3	52.5	63.9

of development. The deep learning detection methods consist of two major part, and the methods mentioned are highly related to the evolution of major pipeline. Finally, the hotspot areas of object detection are described to expect more influential progress in the future. Major part of deep-learning based detectors focus on supervised learning tasks, which employs large amount of training data for the robustness of model. Meanwhile, the direction of using pre-trained models has becoming the growing trend of researches. Using pre-trained models can largely reduce the dependence of abundant amount of training data and labels. It is more applicable in industry and academic regions,

which saves the cost on devices and annotations. The review is helpful for readers to learn the history of object detection and take guidance for future progress. Some future directions and directions are included in following aspects to guide readers to explore more hotpots of object detection.

3-D object detection: With the popularity of 3-D sensors like LiDAR and 3-D cloud points, the collected images becomes closer to human lives (Wang et al., 2022a). The structure of CNN and Transformer has been extended into 3-D visual tasks for large kernel networks. In 2023, Yang et al. (2023) proposed a novel Point-Voxel Transformer named PVT-SSD, which uses sparse convolutions and point voxel module to gain complex context information more efficiently. The input-dependent Query initialization module generates reference points and queries efficiently. It is able to adapt to complex scenes like multiple sensors and overlapping pixels in images. In the same year, Chen et al. (2023) proposed a large-kernel 3D CNN network named LargeKernel3D, which uses spatial-wise partition convolution to design a 3-D large kernel. The spatial-wise partition convolution uses large kernel sizes and shares the same weight with neighbors to simplify object detection performance.

LiDAR point cloud furnishes dependable depth data, which facilitates precise object localization and enabling the characterization of their geometries. 3-D point cloud detection has a promising future for object detection, cross-modal zero-sample recognition of point clouds without any 3-D training can be realized one day (Zhang et al., 2022b; Li et al., 2023a).

Video detection: Real-time object detection is of great importance for autonomous driving and face detection. General object detection methods are mainly based on 2-D images, therefore slicing videos into separate image frames as input of object detection models can be a good idea. However, it is hard to distinguish the slight difference among neighboring areas and the relationship between each frame is hard to detect. In 2023, Hu et al. (2023a) proposed a novel Dynamic Multi-scale Voxel Flow Network(DMVFN) to use distinguishing routing module to understand different scales of video frames. The same year, Gan et al. (2023) introduced a Collaborative noisy Label Cleaner(CLC) module to learn from noisy highlight moments without manual annotation. The method considers noisy labels into detection processes.

Multimodal application: The concept of multi-modality is introduced to deal with cross-modal attention in different representations (Li et al., 2021). Images, videos, sound, text and many other multimedia channels are important to perceive outside world, using multi-modalities to connect information from external is helpful for object detection. In 2022, Zhang et al. (2022a) proposed a novel framework named as CAT-Det, which uses Point former(PT) branch to handle point cloud dataset and Image former(IT) branch along with a cross-modal transformer to deal with image information. After that a One-way Multi-modal Data Augmentation(OMDA) mechanism is employed to improve the accuracy of point cloud detection.

Weakly Supervised object detection: Object detection methods have been widely used in supervised classification and regression tasks. However, it usually requires huge amount of handcrafted labeled data for annotations. Weakly supervised object detection(WSOD) is one general solution to utilize features only from image-level labels rather than object bounding boxes in the whole image (Shao et al., 2021). The WSOD methods can be divided into two main categories: multi-instance learning(MIL) based network and class activation mapping(CAM) network.

The MIL-based network mainly uses multiple supervised learning classifiers in WSOD tasks (Zhang et al., 2020; Shen et al., 2020). Multi-instance learning aims at learning a model from a set of labeled bags that contains several individual instances. For WSOD tasks, MIL can be used to train object detectors where each image is a bag (bag) and regions in the image are instances. The CAM-based network is another branch of WSOD task (Jiang et al., 2021; Zhang et al., 2023). In WSOD tasks, CAM can be used to visualize the activation regions of the model for different categories to help understand the model's decisions and provide information about the location of the detection frame.

Combination of DETR method with CNN-based methods: Although DETR series methods have experienced huge improvement in object detection, CNN-based models have better accuracy in detecting objects. Based on the idea of step-by-step, in recent years, Ouyang (Ouyang, 2022, 2023) combined DETR model with YOLO by using progressive methods to accelerate the training epochs and improve detecting performance. Combining DETR and YOLO models together can improve the quality of information acquisition in former stages, and the last stage use the similar end-to-end structure of DETR without using Non-maximum Suppression(NMS).

Large Language model applications: With the appearance of DETR and ViT series models, the exponential growth on algorithm scales lead to revolutionary improvement on the performance of object detection methods. Researches in exploring large language models have becoming the major trend in the future. In 2023, Zheng et al. (2023) proposed a novel Focus-DETR model to pay more attention to the informative tokens for reconstructing the encoder with dual attention. The same year, Lv et al. (2023) proposed a novel RT-DETR method, which outperforms YOLO series in real time object detection with an efficient hybrid encoder for extracting multi-scale information. The growing trend will still keep a long period.

Generative Adversarial Network(GAN) based detection: In many real-life applications, images are not always clear as natural states, GAN model can be utilized for enhancing the robustness of object detectors. Gupta et al. (2024) utilized several weather augmentation methods to handle images and introduced several denoising methods to enhance the efficiency of these methods. In real results, the training images generated by GAN models should have stronger generalization ability. Combining GAN with object detectors can improve the robustness of models in several extreme conditions like partly covered, blurred or other disturbance (Prakash and Karam, 2021).

CRediT authorship contribution statement

Yibo Sun: Methodology, Writing – original draft. **Zhe Sun:** Conceptualization, Investigation. **Weitong Chen:** Methodology, Resources, Supervision.

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.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant No. 62272239, 62302237).

References

Agarap, Abien Fred, 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

Bell, Sean, Zitnick, C. Lawrence, Bala, Kavita, Girshick, Ross, 2016. 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. pp. 2874–2883.

Bochkovskiy, Alexey, Wang, Chien-Yao, Liao, Hong-Yuan Mark, 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Bolya, Daniel, Zhou, Chong, Xiao, Fanyi, Lee, Yong Jae, 2019. Yolact: Real-time instance segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9157–9166.

Brown, Tom, Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared D., Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, et al., 2020. Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 33, 1877–1901.

Burger, Wilhelm, Burge, Mark J., 2022. Scale-invariant feature transform (SIFT). In: Digital Image Processing: An Algorithmic Introduction. Springer, pp. 709–763.

Burnett, Keenan, Qian, Jingxing, Du, Xintong, Liu, Linqiao, Yoon, David J., Shen, Tianchang, Sun, Susan, Samavi, Sepehr, Sorocky, Michael J., Bianchi, Mollie, et al., 2021. Zeus: A system description of the two-time winner of the collegiate SAE autodrive competition. J. Field Robotics 38 (1), 139–166.

Cai, Zhaowei, Vasconcelos, Nuno, 2018. Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6154–6162.

Carion, Nicolas, Massa, Francisco, Synnaeve, Gabriel, Usunier, Nicolas, Kirillov, Alexander, Zagoruyko, Sergey, 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Caron, Mathilde, Touvron, Hugo, Misra, Ishan, Jégou, Hervé, Mairal, Julien, Bojanowski, Piotr, Joulin, Armand, 2021. Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660.

Cervantes, Jair, Garcia-Lamont, Farid, Rodríguez-Mazahua, Lisbeth, Lopez, Asdrubal, 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing 408, 189–215.

Chen, Liangyu, Chu, Xiaojie, Zhang, Xiangyu, Sun, Jian, 2022. Simple baselines for image restoration. In: European Conference on Computer Vision. Springer, pp. 17–33.

Chen, Yunliang, Joo, Jungseock, 2021. Understanding and mitigating annotation bias in facial expression recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14980–14991.

Chen, Yukang, Liu, Jianhui, Zhang, Xiangyu, Qi, Xiaojuan, Jia, Jiaya, 2023. LargeKernel3D: Scaling up kernels in 3D sparse CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13488–13498.

Chen, Ting, Saxena, Saurabh, Li, Lala, Fleet, David J., Hinton, Geoffrey, 2021. Pix2seq: A language modeling framework for object detection. arXiv preprint arXiv:2109.10852.

Chu, Zihan, 2024. D-YOLO a robust framework for object detection in adverse weather conditions. arXiv preprint arXiv:2403.09233.

Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, Schiele, Bernt, 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223.

Dai, Jifeng, Li, Yi, He, Kaiming, Sun, Jian, 2016. R-fcn: Object detection via region-based fully convolutional networks. Adv. Neural Inf. Process. Syst. 29.

Dai, Jifeng, Qi, Haozhi, Xiong, Yuwen, Li, Yi, Zhang, Guodong, Hu, Han, Wei, Yichen, 2017. Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 764–773.

Dalal, Navneet, Triggs, Bill, 2005. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1. CVPR'05, IEEE, pp. 886–893.

Deng, Xin, Deng, Yufan, Yang, Ren, Yang, Wenzhe, Timofte, Radu, Xu, Mai, 2023. MASIC: Deep mask stereo image compression. IEEE Trans. Circuits Syst. Video Technol.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina, 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Erhan, Dumitru, Szegedy, Christian, Toshev, Alexander, Anguelov, Dragomir, 2014. Scalable object detection using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2154.

Everingham, Mark, Eslami, S.M. Ali, Van Gool, Luc, Williams, Christopher K.I., Winn, John, Zisserman, Andrew, 2015. The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. 111, 98–136.

Everingham, Mark, Van Gool, Luc, Williams, Christopher K.I., Winn, John, Zisserman, Andrew, 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88, 303–338.

Fang, Yuxin, Liao, Bencheng, Wang, Xinggang, Fang, Jiemin, Qi, Jiyang, Wu, Rui, Niu, Jianwei, Liu, Wenyu, 2021. You only look at one sequence: Rethinking transformer in vision through object detection. Adv. Neural Inf. Process. Syst. 34, 26183–26197.

Felzenszwalb, Pedro F., Girshick, Ross B., McAllester, David, 2010. Cascade object detection with deformable part models. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp. 2241–2248.

Fidler, Sanja, Mottaghi, Roozbeh, Yuille, Alan, Urtasun, Raquel, 2013. Bottom-up segmentation for top-down detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3294–3301.

Fu, Cheng-Yang, Liu, Wei, Ranga, Ananth, Tyagi, Ambrish, Berg, Alexander C., 2017. Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659.

Gan, Bei, Shu, Xiujun, Qiao, Ruizhi, Wu, Haoqian, Chen, Keyu, Li, Hanjun, Ren, Bo, 2023. Collaborative noisy label cleaner: Learning scene-aware trailers for multimodal highlight detection in movies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18898–18907.

Ge, Zheng, Liu, Songtao, Wang, Feng, Li, Zeming, Sun, Jian, 2021. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.

Geiger, Andreas, Lenz, Philip, Urtasun, Raquel, 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 3354–3361.

Gidaris, Spyros, Komodakis, Nikos, 2015. Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1134–1142.

Gidaris, Spyros, Komodakis, Nikos, 2016. Locnet: Improving localization accuracy for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 789–798.

Girshick, Ross, 2015. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, Malik, Jitendra, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587.

Grauman, Kristen, Darrell, Trevor, 2005. The pyramid match kernel: Discriminative classification with sets of image features. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Vol. 2. IEEE, pp. 1458–1465.

Guo, Chaoxu, Fan, Bin, Zhang, Qian, Xiang, Shiming, Pan, Chunhong, 2020. Augfpn: Improving multi-scale feature learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12595–12604.

Gupta, Himanshu, Kotlyar, Oleksandr, Andreasson, Henrik, Lilienthal, Achim J., 2024. Robust object detection in challenging weather conditions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7523–7532.

He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, Girshick, Ross, 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. 37 (9), 1904–1916.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George E., Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.

Hinton, Geoffrey E., Salakhutdinov, Ruslan R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507.

Hinton, Geoffrey, Vinyals, Oriol, Dean, Jeff, 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hu, Xiaotao, Huang, Zhewei, Huang, Ailin, Xu, Jun, Zhou, Shuchang, 2023a. A dynamic multi-scale voxel flow network for video prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6121–6131.

Hu, Yihan, Yang, Jiazhi, Chen, Li, Li, Keyu, Sima, Chonghao, Zhu, Xizhou, Chai, Siqi, Du, Senyao, Lin, Tianwei, Wang, Wenhai, et al., 2023b. Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17853–17862.

Huang, Xuehua, Chen, Weihong, Yang, Wangdong, 2021. Improved algorithm based on the deep integration of googlenet and residual neural network. J. Phys. Conf. Ser. 1757 (1), 012069.

Ingle, Palash Yuvraj, Kim, Young-Gab, 2022. Real-time abnormal object detection for video surveillance in smart cities. Sensors 22 (10), 3862.

Ioffe, Sergey, Szegedy, Christian, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. PMLR, pp. 448–456.

Jiang, Peng-Tao, Zhang, Chang-Bin, Hou, Qibin, Cheng, Ming-Ming, Wei, Yunchao, 2021. Layercam: Exploring hierarchical class activation maps for localization. IEEE Trans. Image Process. 30, 5875–5888.

Jiao, Licheng, Zhang, Fan, Liu, Fang, Yang, Shuyuan, Li, Lingling, Feng, Zhixi, Qu, Rong, 2019. A survey of deep learning-based object detection. IEEE Access 7, 128837–128868.

Jocher, Glenn, Chaurasia, Ayush, Qiu, Jing, 2023. Ultralytics YOLO.

Kavukcuoglu, Koray, Sermanet, Pierre, Boureau, Y.-Lan, Gregor, Karol, Mathieu, Michaël, Cun, Yann, et al., 2010. Learning convolutional feature hierarchies for visual recognition. Adv. Neural Inf. Process. Syst. 23.

Khan, Salman, Naseer, Muzammal, Hayat, Munawar, Zamir, Syed Waqas, Khan, Fahad Shahbaz, Shah, Mubarak, 2022. Transformers in vision: A survey. ACM Comput. Surv. 54 (10s), 1–41.

Kong, Tao, Sun, Fuchun, Tan, Chuanqi, Liu, Huaping, Huang, Wenbing, 2018. Deep feature pyramid reconfiguration for object detection. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 169–185.

Kong, Tao, Sun, Fuchun, Yao, Anbang, Liu, Huaping, Lu, Ming, Chen, Yurong, 2017. Ron: Reverse connection with objectness prior networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5936–5944.

Kong, Tao, Yao, Anbang, Chen, Yurong, Sun, Fuchun, 2016. Hypernet: Towards accurate region proposal generation and joint object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 845–853.

Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey E., 2012. Imagenet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. 25.

Kumar, Debasis, Muhammad, Naveed, 2023. Object detection in adverse weather for autonomous driving through data merging and YOLOv8. Sensors 23 (20), 8471.

Kuo, Weicheng, Hariharan, Bharath, Malik, Jitendra, 2015. Deepbox: Learning objectness with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2479–2487.

Law, Hei, Deng, Jia, 2018. Cornernet: Detecting objects as paired keypoints. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 734–750.

Lazebnik, Svetlana, Schmid, Cordelia, Ponce, Jean, 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2. CVPR'06, IEEE, pp. 2169–2178.

Li, Zeming, Chen, Yilun, Yu, Gang, Deng, Yangdong, 2018a. R-fcn++: Towards accurate region-based fully convolutional networks for object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, No. 1.

Li, Shanshan, Gao, Pan, Tan, Xiaoyang, Wei, Mingqiang, 2023a. ProxyFormer: Proxy alignment assisted point cloud completion with missing part sensitive transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9466–9475.

Li, Chuyi, Li, Lulu, Jiang, Hongliang, Weng, Kaiheng, Geng, Yifei, Li, Liang, Ke, Zaidan, Li, Qingyuan, Cheng, Meng, Nie, Weiqiang, et al., 2022a. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

Li, Yaping, Li, Aifeng, Li, Xiaoyu, Liang, Dongyue, 2022b. Detection and identification of peach leaf diseases based on YOLO v5 improved model. In: Proceedings of the 5th International Conference on Control and Computer Vision. pp. 79–84.

Li, Zhaoxin, Lu, Shuhua, Dong, Yishan, Guo, Jingyuan, 2023b. Msffa: a multi-scale feature fusion and attention mechanism network for crowd counting. Vis. Comput. 39 (3), 1045–1056.

Li, Yanghao, Mao, Hanzi, Girshick, Ross, He, Kaiming, 2022c. Exploring plain vision transformer backbones for object detection. In: European Conference on Computer Vision. Springer, pp. 280–296.

Li, Zeming, Peng, Chao, Yu, Gang, Zhang, Xiangyu, Deng, Yangdong, Sun, Jian, 2018b. Detnet: Design backbone for object detection. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 334–350.

Li, Junnan, Selvaraju, Ramprasaath, Gotmare, Akhilesh, Joty, Shafiq, Xiong, Caiming, Hoi, Steven Chu Hong, 2021. Align before fuse: Vision and language representation learning with momentum distillation. Adv. Neural Inf. Process. Syst. 34, 9694–9705.

Li, Yuancheng, Zhou, Shenglong, Chen, Hui, 2022d. Attention-based fusion factor in FPN for object detection. Appl. Intell. 52 (13), 15547–15556.

Liang, Ming, Hu, Xiaolin, 2015. Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3367–3375.

Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, Belongie, Serge, 2017a. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, Dollár, Piotr, 2017b. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, Zitnick, C. Lawrence, 2014. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13. Springer, pp. 740–755.

Liu, Wei, Anguelov, Dragomir, Erhan, Dumitru, Szegedy, Christian, Reed, Scott, Fu, Cheng-Yang, Berg, Alexander C., 2016. Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, pp. 21–37.

Liu, Shu, Qi, Lu, Qin, Haifang, Shi, Jianping, Jia, Jiaya, 2018. Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8759–8768.

Lu, Xin, Li, Buyu, Yue, Yuxin, Li, Quanquan, Yan, Junjie, 2019. Grid r-cnn. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7363–7372.

Lv, Wenyu, Xu, Shangliang, Zhao, Yian, Wang, Guanzhong, Wei, Jinman, Cui, Cheng, Du, Yuning, Dang, Qingqing, Liu, Yi, 2023. Detrs beat yolos on real-time object detection. arXiv preprint arXiv:2304.08069.

Ma, Chunyan, Li, Xin, Li, Yujie, Tian, Xinliang, Wang, Yichuan, Kim, Hyoungseop, Serikawa, Seiichi, 2021. Visual information processing for deep-sea visual monitoring system. Cogn. Robotics 1, 3–11.

Maninis, Kevis-Kokitsi, Caelles, Sergi, Pont-Tuset, Jordi, Van Gool, Luc, 2018. Deep extreme cut: From extreme points to object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 616–625.

Misra, Diganta, 2019. Mish: A self regularized non-monotonic activation function. arXiv preprint arXiv:1908.08681.

Mo, Kaichun, Zhu, Shilin, Chang, Angel X, Yi, Li, Tripathi, Subarna, Guibas, Leonidas J., Su, Hao, 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 909–918.

O. Pinheiro, Pedro O., Collobert, Ronan, Dollár, Piotr, 2015. Learning to segment object candidates. Adv. Neural Inf. Process. Syst. 28.

Ouyang, Haodong, 2022. DEYO: DETR with YOLO for step-by-step object detection. arXiv preprint arXiv:2211.06588.

Ouyang, Haodong, 2023. DEYOv2: Rank feature with greedy matching for end-to-end object detection. arXiv preprint arXiv:2306.09165.

Ouyang, Wanli, Wang, Xiaogang, Zeng, Xingyu, Qiu, Shi, Luo, Ping, Tian, Yonglong, Li, Hongsheng, Yang, Shuo, Wang, Zhe, Loy, Chen-Change, et al., 2015. Deepid-net: Deformable deep convolutional neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2403–2412.

Pont-Tuset, Jordi, Arbelaez, Pablo, Barron, Jonathan T., Marques, Ferran, Malik, Jitendra, 2016. Multiscale combinatorial grouping for image segmentation and object proposal generation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (1), 128–140.

Prakash, Charan D., Karam, Lina J., 2021. It GAN DO better: GAN-based detection of objects on images with varying quality. IEEE Trans. Image Process. 30, 9220–9230.

Qi, Delong, Tan, Weijun, Yao, Qi, Liu, Jingfeng, 2022. YOLO5Face: why reinventing a face detector. In: European Conference on Computer Vision. Springer, pp. 228–244.

Radford, Alec, Narasimhan, Karthik, Salimans, Tim, Sutskever, Ilya, et al., 2018. Improving Language Understanding by Generative Pre-Training. OpenAI.

Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, Sutskever, Ilya, et al., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, Liu, Peter J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. 21 (1), 5485–5551.

Rani, Shreenagamanjula, Ramani, D. Roja, Raj, V. Ravi, Ujwal, V., Nk, Sacheth, 2023. A deep learning model for collective disorder using visual geometry group 16. In: 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems. ICAECIS, IEEE, pp. 594–599.

Redmon, Joseph, Divvala, Santosh, Girshick, Ross, Farhadi, Ali, 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Redmon, Joseph, Farhadi, Ali, 2017. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7263–7271.

Redmon, Joseph, Farhadi, Ali, 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, Sun, Jian, 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Roh, Byungseok, Shin, JaeWoong, Shin, Wuhyun, Kim, Saehoon, 2021. Sparse detr: Efficient end-to-end object detection with learnable sparsity. arXiv preprint arXiv:2111.14330.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115, 211–252.

Shao, Feifei, Chen, Long, Shao, Jian, Ji, Wei, Xiao, Shaoning, Ye, Lu, Zhuang, Yueting, Xiao, Jun, 2021. Deep learning for weakly-supervised object detection and object localization: a survey. arXiv preprint arXiv:2105.12694.

Shen, Yunhang, Ji, Rongrong, Chen, Zhiwei, Wu, Yongjian, Huang, Feiyue, 2020. UWSOD: Toward fully-supervised-level capacity weakly supervised object detection. Adv. Neural Inf. Process. Syst. 33, 7005–7019.

Shepley, Andrew J., Falzon, Greg, Kwan, Paul, Brankovic, Ljiljana, 2023. Confluence: A robust non-iou alternative to non-maxima suppression in object detection. IEEE Trans. Pattern Anal. Mach. Intell.

Shrivastava, Abhinav, Gupta, Abhinav, Girshick, Ross, 2016. Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 761–769.

Simon, Martin, Amende, Karl, Kraus, Andrea, Honer, Jens, Samann, Timo, Kaulbersch, Hauke, Milz, Stefan, Michael Gross, Horst, 2019. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, Rabinovich, Andrew, 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.

Touvron, Hugo, Cord, Matthieu, Douze, Matthijs, Massa, Francisco, Sablayrolles, Alexandre, Jégou, Hervé, 2021a. Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. PMLR, pp. 10347–10357.

Touvron, Hugo, Cord, Matthieu, Sablayrolles, Alexandre, Synnaeve, Gabriel, Jégou, Hervé, 2021b. Going deeper with image transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 32–42.

Uijlings, Jasper R.R., Van De Sande, Koen E.A., Gevers, Theo, Smeulders, Arnold W.M., 2013. Selective search for object recognition. Int. J. Comput. Vis. 104, 154–171.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Łukasz, Polosukhin, Illia, 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, Chien-Yao, Bochkovskiy, Alexey, Liao, Hong-Yuan Mark, 2021a. Scaled-yolov4: Scaling cross stage partial network. In: Proceedings of the IEEE/Cvf Conference on Computer Vision and Pattern Recognition. pp. 13029–13038.

Wang, Chien-Yao, Bochkovskiy, Alexey, Liao, Hong-Yuan Mark, 2023. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7464–7475.

Wang, Yue, Guizilini, Vitor Campagnolo, Zhang, Tianyuan, Wang, Yilun, Zhao, Hang, Solomon, Justin, 2022a. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. PMLR, pp. 180–191.

Wang, Tao, Yuan, Li, Chen, Yunpeng, Feng, Jiashi, Yan, Shuicheng, 2021b. Pnp-detr: Towards efficient visual analysis with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4661–4670.

Wang, Yingming, Zhang, Xiangyu, Yang, Tong, Sun, Jian, 2022b. Anchor detr: Query design for transformer-based detector. In: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 3. pp. 2567–2575.

Wu, Xiongwei, Sahoo, Doyen, Hoi, Steven C.H., 2020. Recent advances in deep learning for object detection. Neurocomputing 396, 39–64.

Wu, Haiping, Xiao, Bin, Codella, Noel, Liu, Mengchen, Dai, Xiyang, Yuan, Lu, Zhang, Lei, 2021. Cvt: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22–31.

Yang, Honghui, Wang, Wenxiao, Chen, Minghao, Lin, Binbin, He, Tong, Chen, Hua, He, Xiaofei, Ouyang, Wanli, 2023. PVT-ssd: Single-stage 3D object detector with point-voxel transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13476–13487.

Yao, Zhuyu, Ai, Jiangbo, Li, Boxun, Zhang, Chi, 2021. Efficient detr: improving end-to-end object detector with dense prior. arXiv preprint arXiv:2104.01318.

Zeiler, Matthew D., Fergus, Rob, 2014. Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13. Springer, pp. 818–833.

Zhang, Yanan, Chen, Jiaxin, Huang, Di, 2022a. Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 908–917.

Zhang, Renrui, Guo, Ziyu, Zhang, Wei, Li, Kunchang, Miao, Xupeng, Cui, Bin, Qiao, Yu, Gao, Peng, Li, Hongsheng, 2022b. Pointclip: Point cloud understanding by clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8552–8562.

Zhang, Shifeng, Wen, Longyin, Bian, Xiao, Lei, Zhen, Li, Stan Z., 2018. Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4203–4212.

Zhang, Shengchuan, Yu, Songlin, Ding, Haixin, Hu, Jie, Cao, Liujuan, 2023. CAM R-CNN: End-to-end object detection with class activation maps. Neural Process. Lett. 1–17.

Zhang, Dingwen, Zeng, Wenyuan, Yao, Jieru, Han, Junwei, 2020. Weakly supervised object detection using proposal-and semantic-level relationships. IEEE Trans. Pattern Anal. Mach. Intell. 44 (6), 3349–3363.

Zhang, Yongbing, Zhao, Debin, Zhang, Jian, Xiong, Ruiqin, Gao, Wen, 2011. Interpolation-dependent image downsampling. IEEE Trans. Image Process. 20 (11), 3291–3296.

Zhao, Gangming, Ge, Weifeng, Yu, Yizhou, 2021. Graphfpn: Graph feature pyramid network for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2763–2772.

Zhao, Zhong-Qiu, Zheng, Peng, Xu, Shou-tao, Wu, Xindong, 2019. Object detection with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 30 (11), 3212–3232.

Zheng, Dehua, Dong, Wenhui, Hu, Hailin, Chen, Xinghao, Wang, Yunhe, 2023. Less is more: Focus attention for efficient detr. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6674–6683.

Zheng, Minghang, Gao, Peng, Zhang, Renrui, Li, Kunchang, Wang, Xiaogang, Li, Hongsheng, Dong, Hao, 2020. End-to-end object detection with adaptive clustering transformer. arXiv preprint arXiv:2011.09315.

Zhong, Zhuoyao, Sun, Lei, Huo, Qiang, 2019. An anchor-free region proposal network for faster R-CNN-based text detection approaches. Int. J. Document Anal. Recognit. 22, 315–327.

Zhou, Peng, Ni, Bingbing, Geng, Cong, Hu, Jianguo, Xu, Yi, 2018. Scale-transferrable object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 528–537.

Zhou, Xingyi, Wang, Dequan, Krähenbühl, Philipp, 2019a. Objects as points. arXiv preprint arXiv:1904.07850.

Zhou, Xingyi, Zhuo, Jiacheng, Krahenbuhl, Philipp, 2019b. Bottom-up object detection by grouping extreme and center points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 850–859.

Zhu, Xizhou, Hu, Han, Lin, Stephen, Dai, Jifeng, 2019. Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9308–9316.

Zhu, Linxiang, Lee, Feifei, Cai, Jiawei, Yu, Hongliu, Chen, Qiu, 2022. An improved feature pyramid network for object detection. Neurocomputing 483, 127–139.

Zhu, Xizhou, Su, Weijie, Lu, Lewei, Li, Bin, Wang, Xiaogang, Dai, Jifeng, 2020. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159.

Zhu, Yousong, Zhao, Chaoyang, Wang, Jinqiao, Zhao, Xu, Wu, Yi, Lu, Hanqing, 2017. Couplenet: Coupling global structure with local parts for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4126–4134.

Zou, Zhengxia, Chen, Keyan, Shi, Zhenwei, Guo, Yuhong, Ye, Jieping, 2023. Object detection in 20 years: A survey. Proc. IEEE 111 (3), 257–276.