Engineering Applications of Artificial Intelligence 126 (2023) 107021

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Survey paper

Transformer for object detection: Review and benchmark ☆

Yong Li [a,*], Naipeng Miao [a], Liangdi Ma [b], Feng Shuang [a], Xingwen Huang [a]

[a] Guangxi Key Laboratory of Intelligent Control and Maintenance of Power Equipment, School of Electrical Engineering, Guangxi University, No. 100, Daxuedong Road, Xixiangtang District, Nanning, 530004, Guangxi, China

[b] School of Software, Tsinghua University, No. 30 Shuangqing Road, Haidian District, Beijing, 100084, China

ARTICLE INFO

Keywords:
Review
Object detection
Transformer-based models
COCO2017 dataset
Benchmark

ABSTRACT

Object detection is a crucial task in computer vision (CV). With the rapid advancement of Transformer-based models in natural language processing (NLP) and various visual tasks, Transformer structures are becoming increasingly prevalent in CV tasks. In recent years, numerous Transformer-based object detectors have been proposed, achieving performance comparable to mainstream convolutional neural network-based (CNN-based) approaches. To provide researchers with a comprehensive understanding of the development, advantages, disadvantages, and future potential of Transformer-based object detectors in Artificial Intelligence (AI), this paper systematically reviews the mainstream methods and analyzes the limitations and challenges encountered in their current applications, while also offering insights into future research directions. We have reviewed a large number of papers, selected the most prominent Transformer detection methods, and divided them into Transformer Neck and Transformer Backbone categories for introduction and comparative analysis. Furthermore, we have constructed a benchmark using the COCO2017 dataset to evaluate different object detection algorithms. Finally, we summarize the challenges and prospects in this field.

1. Introduction

Object detection is a fundamental task in computer vision that requires simultaneous classification and localization of potential objects within a single image (Zhao et al., 2019). As such, it plays a crucial role in various applications, including autonomous driving (Chen et al., 2015, 2017), face recognition (Sung and Poggio, 1998), pedestrian detection (Dollar et al., 2012), and medical detection (Kobatake and Yoshinaga, 1996). The performance of object detection directly influences object tracking, environment perception, and scene understanding (Felzenszwalb et al., 2010). Recently, deep learning-based object detection methods have gained considerable attention due to the rapid development of deep learning. However, numerous challenges remain, such as balancing accuracy and efficiency, handling multi-scale objects, and creating lightweight models.

Traditional mainstream object detection methods have predominantly utilized convolutional neural networks (CNNs), including Faster R-CNN (Ren et al., 2016), SSD (Liu et al., 2016), and YOLO with its variants (Redmon et al., 2016; Redmon and Farhadi, 2018, 2017; Bochkovskiy et al., 2020; Ge et al., 2021). Owing to the remarkable success of Transformers in natural language processing (NLP), researchers have endeavored to adapt Transformer architectures for computer vision tasks. As a result, numerous Transformer-based vision models have emerged in recent years, achieving performance levels that are comparable or even superior to their CNN counterparts.

Transformer (Vaswani et al., 2017) was initially proposed as an architecture based on the self-attention mechanism for machine translation and sequence modeling tasks (Sutskever et al., 2014). In recent years, Transformer has experienced significant advancements in NLP and has become a mainstream deep learning model, such as BERT (Devlin et al., 2018) and its variants (Lan et al., 2019; Liu et al., 2019), GPT series (Radford et al., 2018, 2019; Brown et al., 2020), and others. Due to its scalability, Transformer can be pre-trained on large datasets and subsequently fine-tuned for downstream tasks.

Transformers in object detection have garnered increasing attention, particularly over the last three years. Several high-performance models have been proposed, such as DETR (Carion et al., 2020), Deformable DETR (Dai et al., 2017), Swin Transformer (Liu et al., 2021b,a), DINO (Zhang et al., 2022a), and more. Currently, Transformer-based models have emerged as a new paradigm in object detection, making a systematic analysis and evaluation of numerous existing Transformer-based detectors essential for future research.

☆ This work was supported by the Guangxi Science and Technology base and Talent Project (Grant No. Guike AD22080043), the Key Laboratory of Advanced Manufacturing Technology, Ministry of Education (Grant No. GZUAMT2021KF04), and the National Natural Science Foundation of China (Grant No. 61720106009).

* Corresponding author.
E-mail address: yongli@gxu.edu.cn (Y. Li).

https://doi.org/10.1016/j.engappai.2023.107021
Received 22 October 2022; Received in revised form 25 May 2023; Accepted 19 August 2023
Available online 4 September 2023

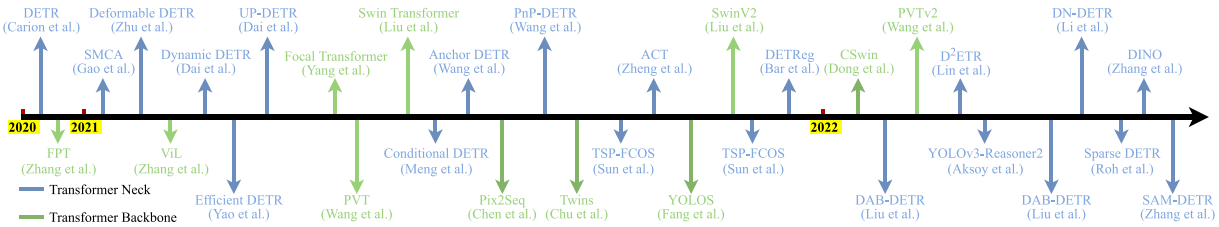


Fig. 1. Chronological overview of most Transformer-based object detection methods.

Some reviews (Khan et al., 2021; Liu et al., 2021c; Arkin et al., 2021; Han et al., 2022; Arkin et al., 2022) have provided detailed introductions and analyses of Transformer-based detectors. In contrast to these surveys, our study not only presents a thorough comparison of the strengths and weaknesses of object detectors based on both Transformer and CNN architectures, but also classifies the prevalent Transformer-based detectors into Transformer Backbone and Transformer Neck categories. Moreover, we systematically analyze their performance, potential, and limitations. We investigate the advancements and constraints of various state-of-the-art Transformer-based detectors (Table 6) and establish benchmarks for these methods using the COCO2017 dataset (Tables. 4, 5). We hope this review delivers a comprehensive understanding of Transformer-based object detectors for researchers.

We have categorized existing methods into two groups based on the role of Transformer in the overall model architecture: Transformer Neck and Transformer Backbone, as illustrated in Fig. 1. We present a detailed analysis of representative methods, compare these methods horizontally on the COCO2017 dataset (Lin et al., 2014), and summarize the novelty of each method, such as Transformer Backbone with hierarchical representation structure, spatial prior acceleration based on sparse attention, and pure sequence processing for object detection, among others. The main contributions of this paper are as follows:

1. We provide a comprehensive summary of state-of-the-art Transformer-based object detectors from the past three years, highlighting recent breakthroughs in Transformer architecture for object detection. For each representative model, we offer an in-depth analysis while examining its relationship and connections with other models, both incrementally and comparatively. Moreover, we compare the strengths and weaknesses of Transformer and CNN architectures, and further discuss the performance, key features, and limitations of both Transformer Neck (DETR-like models) and Transformer Backbone (ViT-like models).
2. We comprehensively compare mainstream models on the same dataset, establish a benchmark based on the COCO2017 dataset, and offer insightful discussions.
3. We present an in-depth analysis of the transition as Transformer architecture extends from sequence to visual tasks. Furthermore, we discuss the future development of Transformer and CNN approaches in object detection.

The rest structure of this paper is organized as follows. Section 2 introduces the main object detection datasets and evaluation metrics, as well as the Attention mechanism and Transformer basic architecture. Section 3 outlines the current mainstream Transformer-based object detectors. Section 4 discusses the methods of these models in a multi-level comparison. Section 5 concludes the paper with an outlook.

2. Transformer architecture

Transformer is an architecture based on the attention mechanism proposed by Vaswani et al. (2017) in 2017, which was initially used for machine translation tasks and subsequently achieved great success in NLP (Devlin et al., 2018). The success of Transformer is attributed to its unique architecture, whose core design is the Encoder–Decoder structure based on self-attention. As shown in Fig. 2, Transformer consists of three main blocks: multi-headed attention, positional encoding, and feed-forward network. Multi-head attention (MHA) block and Feed-forward network block are the main modules of Encoder and Decoder. Position encoding is a vital module to all Transformer variants and is responsible for attaching position information to the input sequence. In this section, these fundamental techniques are described in detail.

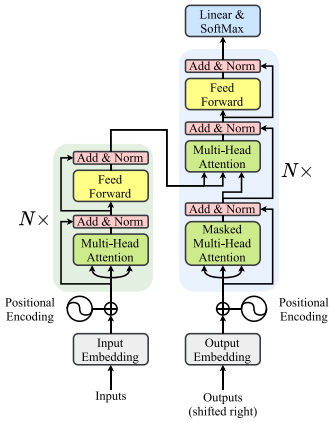


Fig. 2. Transformer structure. The Input Embedding module of Transformer encoder (left column) can map the input sequence to Embedding space and pass it to encoder module for processing. The Transformer decoder (right column) receives the previous output sequence and the output sequence from the intermediate encoder. The previous output sequence will be shifted one bit to the right, and the start token will be appended before the sequence to get the input from the decoder. The feed-forward network and the multi-head attention module are repeated N times to form the encoder and decoder.

2.1. Basic architecture

The structure of Transformer is based on encoder–decoder. The encoder consists of N basic encoder modules, as shown in Fig. 2. Every encoder module consists of a multi-head attention module (MHA) and a feed-forward network (FFN). And then, they are cascaded with residual connection and layer normalization one by one. Finally, the output of the encoder module is shown in Eq. (1):

$$Output = LayerNorm(x + SubLayer(x)), \tag{1}$$

where x is the input sequence, and $SubLayer$ represents the attention module or feedforward network.

2.2. Self-attention

2.2.1. Scaled dot-product attention

The self-attention mechanism module, as the core component of Transformer, consists of two main parts: (1) Linear projection layer: the input sequence is mapped into 3 different vectors (query Q, key K, value V). The input sequences are $X \in \mathbb{R}^{n_x \times d_x}$ and $Y \in \mathbb{R}^{n_y \times d_y}$,

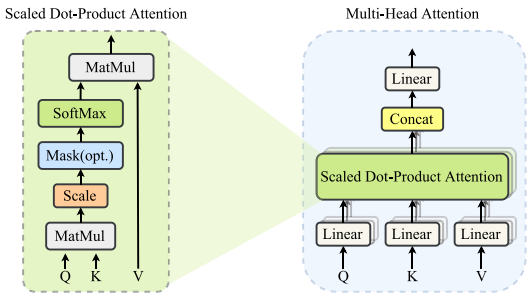


Fig. 3. (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

where n and d denote the length and dimension of the input sequence, respectively. Then Q, K, and V are generated as follows:

$$Q = XW^Q, \quad K = YW^K, \quad V = YW^V, \tag{2}$$

where $W^Q \in \mathbb{R}^{d_x \times d^q}$, $W^K \in \mathbb{R}^{d_y \times d^k}$ and $W^V \in \mathbb{R}^{d_y \times d^v}$ are the learnable weight matrices. The d^q and d^k denotes the dimensions of W^Q and W^K, respectively. The dimension of W^V is d^v. When $X = Y$, Eq. (2) is the self-attention computation, and when $X \neq Y$, it is the cross-attention computation in the Decoder module.

(2) Attention layer: Transformer adopts a special attention method called Scaled Dot-Product Attention, as shown in Fig. 3 (left). The input consists of Q in d^q dimensions, K in d^k dimensions and V in d^v dimensions, and the scaled attention matrix is calculated as shown in Eq. (3).

$$\text{Attention}\ (Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3}$$

where $\frac{1}{\sqrt{d_k}}$ is the scaling factor. The attention weights are obtained by computing the dot product of Q for all K. The attention weights are then normalized by the scaling factor $\frac{1}{\sqrt{d_k}}$ and the softmax layer. The output weights are assigned to the corresponding elements of V to obtain the final attention matrix.

2.2.2. Multi-head attention

However, the modeling ability of single-head attention is weak. To address this problem, Vaswani et al. (2017) proposed multi-head attention (MHA). The structure is shown in Fig. 3 (right). MHA can enhance the modeling ability of each attention layer without changing the number of parameters.

Compared to single-head attention, MHA maps Q, K, and V linearly to different dimensional subspaces (d_q, d_k, d_v) to compute similarity and compute the attention function in parallel. As shown in Eq. (4), the resulting vectors are concatenated and mapped again to obtain the final output.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}\left(\text{head}_1, \ldots, \text{head}_h\right) W^O, \\ \text{where head}_i &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right), \end{aligned} \tag{4}$$

where $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d^q}, W_i^K \in \mathbb{R}^{d_{\text{model}} \times d^k}, W_i^V \in \mathbb{R}^{d_{\text{model}}} \times d^v, W_i^O \in \mathbb{R}^{d_{\text{model}} \times hd^v}$ is the projection parameter matrix. Multi-head attention reduces the dimensionality of each vector when calculating the attention of each head, which reduces overfitting to a certain extent. Since attention has different distributions in different subspaces, this module fuses the feature relationships between different sequence dimensions in vector concatenation.

2.3. Position-wise feed-forward networks

The output of the MHA layer is fed into the feed-forward network (FFN). FFN is mainly composed of two linear transformations with a RuLU activation in between. The output of FFN can be expressed as shown in Eq. (5):

$$\text{FFN}(x) = \max\left(0, xW_1 + b_1\right)W_2 + b_2, \tag{5}$$

where W_1 and W_2 denote weight matrices of the two fully connected layers.

2.4. Positional encoding

Unlike CNN and RNN, self-attention computation brings the advantage of parallel computing while losing word order information. Therefore, positional encoding is used to provide positional information to the model. In detail, a position-dependent signal is added to each word embedding for each input sequence to help the model incorporate the order of words. The output of positional encoding has the same dimension as the embedding layer. So it can be superimposed directly on Embedding. The positional information of each token (a sequence of primitives obtained after the text has been divided into words) and its semantic information (Embedding) are fully integrated and passed to the subsequent layer.

There are many variants of positional encoding. The original Transformer uses sine and cosine functions for positional encoding, as shown in Eq. (6).

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{\text{model}}}\right), \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{\text{model}}}\right), \end{aligned} \tag{6}$$

where pos is the position and i is the dimension. That is, each dimension of the position encoding corresponds to a sine wave. The wavelengths form a geometric progression from 2π to $10000 \times 2\pi$. For any fixed offset k, PE_{pos+k} can be represented as a linear function of PE_{pos}.

2.5. Remarks

The self-attention mechanism allows Transformer to break through the limitation that RNN models cannot be computed in parallel and improve the computational efficiency. Compared with CNN, the self-attentive mechanism has a global perceptual field. The number of operations required to compute the association between two locations does not grow with distance, so it has a stronger ability to learn long-range dependencies. In addition, Transformer has a general modeling capability. Transformer can be regarded as a fully connected graph modeling method that can model heterogeneous nodes by projecting them into a comparable space to compute similarity. Therefore, there is a sufficient theoretical basis for using Transformer for various computer vision tasks based on its general modeling capability. Considering the dimensional differences between images and text, the images are converted into sequences and can then be input into the model for processing.

Moreover, we compare the characteristics of CNN and Transformer. As shown in Table 1, Transformer tends to model shapes more but requires massive data for training. In contrast, CNN tends to model local textures more but has to pile many convolutional layers to have a large enough receptive field to get global information (Geirhos et al., 2019).

3. Transformer for object detection

This section first introduces common datasets and evaluation metrics for object detection and analyzes classic Transformer-based object detectors. According to their structural difference, We classify the listed detectors as Transformer Neck-based detectors and Transformer Backbone-based detectors. The Transformer Neck-based detector infers the class labels and bounding box coordinates with a set of learnable object queries but does not change the backbone used for feature extraction. Transformer Backbone-based detectors propose a generic visual backbone that flattens the image into a sequence instead of convolution for feature extraction. Multiscale feature fusion is also

Table 1
Summary of the highlights and limitations of CNN and Transformer.

Architecture	Highlights	Limitations
Transformer	**(1)** The attention mechanism amplifies the significance of crucial aspects of an image while reducing the rest, thereby concentrating on more relevant features. This mechanism assists the Transformer in modeling the long-range dependencies of input sequence elements and thus enhances its generalization ability for samples outside the distribution (Bai et al., 2021). **(2)** Unlike methods such as RNN and LSTM, the Transformer allows for parallel computations. **(3)** Given its straightforward yet adaptable design, the Transformer can tackle multiple tasks simultaneously, rendering it a potential candidate for a general-purpose model handling various tasks.	**(1)** Transformer-based models are known for their substantial data requirements and computationally expensive nature, particularly when applied to vision tasks (He et al., 2021). **(2)** They are also characterized by a slower rate of convergence, which can pose challenges in their utilization (Gao et al., 2021). **(3)** Further, these models often involve high computational overhead, which exacerbates their deployment issues in resource-constrained settings (Li et al., 2022).
CNN	**(1)** CNN-based models have strong local feature extraction ability benefited from inductive bias properties such as translation invariance, weight sharing, and sparse connectivity. **(2)** CNNs can operate in parallel with lower computational complexity than Transformer.	**(1)** CNN rarely encodes relative feature positions, instead favoring receptive field expansion via larger kernels or stacked layers, often reducing local convolution's computational and statistical efficiency. **(2)** CNN's global feature capture is comparatively weaker than Transformer models (Liu et al., 2021b).

Table 2
Briefing on datasets for object detection.

Name	Image volume	class	Source	Annotation format
VOC2007	9963	20	PASCAL	XML
VOC2012	17112	20	PASCAL	XML
COCO2017	121408	80	Microsoft	JSON

incorporated in many methods to improve detection accuracy and replace the CNN backbone in classical detectors. In reviewing these methods, we summarize the optimization innovations or modules of the different methods. Finally, we compare their performance in Table 4 and Table 5 and give analyzation and discussion on improvements of the above methods.

3.1. Common datasets and evaluation metrics

3.1.1. Common datasets for object detection

Datasets are the basis for measuring and comparing algorithm performance. The commonly used object detection datasets are Pascal VOC2007(Everingham et al., 2007), Pascal VOC2012(Everingham et al., 2012) and Microsoft COCO2017(Lin et al., 2014), as shown in Table 2. The Pascal VOC dataset has only 20 object categories and is regarded as a benchmark dataset for object detection. Compared with VOC, the COCO dataset has more small objects and more objects in a single image, and most of the objects are non-centrally distributed and more similar to the real environment. Thus COCO dataset is more difficult for object detection and has been the mainstream object detection dataset in recent years.

3.1.2. Evaluation metrics

Common evaluation metrics for object detection include Precision, Recall, Average Precision (AP), and mean Average Precision (mAP). In addition to classification, the object detection task localizes the object further with a bounding box associated with its corresponding confidence score to report how certain the bounding box of the object class is detected. Therefore to determine how many objects were detected correctly and how many false positives were generated, we use the Intersection over Union (IoU) metric.

Intersection over Union (IoU). IoU is an evaluation metric that quantifies the similarity between the ground truth bounding box ($gt\ box$) and the predicted bounding box ($pd\ box$) to evaluate how good the predicted box is. The IoU score ranges from 0 to 1; the closer the two boxes, the higher the IoU score. It can be calculated as follow:

$$IoU(gt, pd) = \frac{\text{area}(gt\ box \cap pd\ box)}{\text{area}(gt\ box \cup pd\ box)}, \tag{7}$$

For the IoU threshold at α, True Positive(TP) is a detection for which $IoU(gt, pd) \geq \alpha$ and False Positive (FP) is a detection for which $IoU(gt, pd) \leq \alpha$. False Negative (FN) is a ground-truth missed together with gt for which $IoU(gt, pd) \leq \alpha$. The definitions of TP, TN, FP and FN are shown in Table 3.

Precision. Precision is the probability of the predicted bounding boxes matching actual ground truth boxes, also referred to as the positive predictive value. Precision scores range from 0 to 1, with a high precision implying that most detected objects match ground truth objects.

Recall. Recall is the true positive rate, also referred to as sensitivity, which measures the probability of ground truth objects being correctly detected. Similarly, Recall ranges from 0 to 1, where a high recall score means that most ground truth objects were detected.

The Precision and Recall can be calculated as follow:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{9}$$

Average precision (AP). AP is Area Under the Precision–Recall Curve evaluated at a specific IoU threshold. AP is a single number metric that combines precision and recall and describes the Accuracy–Recall curve by AP among recall values ranging from 0 to 1. It is used to evaluate the performance of object detectors.

Table 3
Definition of terms.

Terms	Definitions
TP (True Positive)	Positive samples are correctly identified as positive samples.
TN (True Negative)	Negative samples are correctly identified as negative samples.
FP (False Positive)	False positive samples, that is, negative samples are mistakenly identified as positive samples.
FN (False Negative)	False negative samples, that is, positive samples are wrongly identified as negative samples.

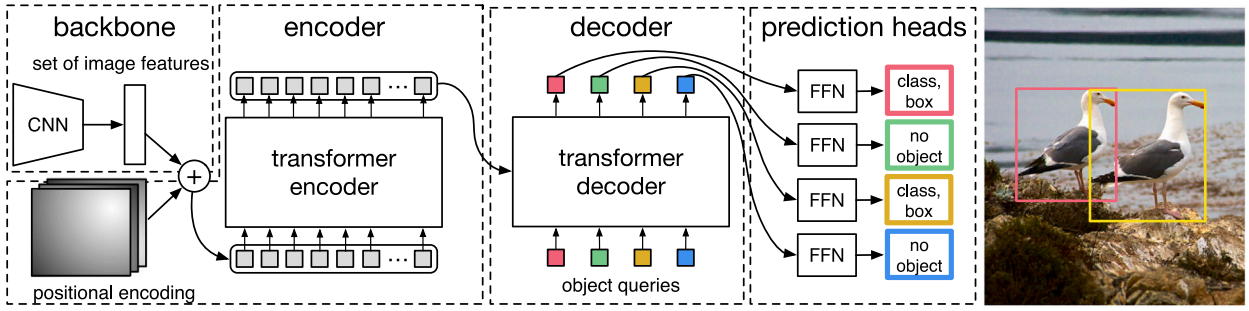


Fig. 4. The pipeline of DETR. The backbone is a convolutional neural network (CNN) that serves as a feature extractor. And Transformer is the core of the DETR architecture, consisting of an encoder and a decoder. The high-dimensional feature map from the backbone is flattened and fed into the encoder. Then encoder processes the spatial information and outputs a sequence of encoded feature vectors. Finally, The output of the decoder is passed through a series of linear layers to predict the final bounding box coordinates and class probabilities for each object query.
Source: Image from Carion et al. (2020).

Mean average precision (mAP). AP is calculated for each class individually, and mAP is the average of AP values across all classes. The mAP can be calculated as Eq. (10). There are two kinds of mAPs commonly used. (1) PASCAL VOC challenge uses mAP as a metric with an IoU threshold of 0.5. (2) While MS COCO averages mAP over different IoU thresholds 50% to 95% with a step of 0.05, this metric is denoted in papers by mAP@[.5,.95]. Therefore, COCO not only averages AP over all classes but also on the defined IoU thresholds.

$$\text{mAP} = \frac{\sum_{i=1}^{k} \text{AP}_i}{k} \text{ for } k \text{ classes}, \tag{10}$$

Frame Per Second (FPS). FPS defines how fast your object detection model processes your video and generates the desired output.

3.2. Transformer neck

In this section, we review the classic Transformer Neck-based object detection models in last two years, starting from the original Transformer detector DETR (Carion et al., 2020). The original DETR regards object detection as end-to-end set prediction, thus removing hand-designed components such as anchor boxes and non-maximum suppression (NMS). However, some drawbacks need to be solved in DETR, such as slow convergence and poor detection of small objects. Therefore, many approaches (sparse attention, spatial prior acceleration, multi-scale detection) have been proposed to improve it by researchers. We compare the performance of all methods together on the COCO2017 dataset with the benchmark shown in Table 4.

3.2.1. DETR

DETR proposed by Carion et al. (2020) is the first object detector that successfully uses the Transformer as the main module in object detection. DETR not only has a simpler and more flexible structure but also has comparable performance compared to previous SOTA approaches, such as the highly optimized Faster R-CNN. Unlike classical object detectors, DETR is an end-to-end object detection model. It gets rid of the autoregressive model, performs parallel inference on object relationships and global image context, and then outputs the final predictions. The structure of DETR is shown in Fig. 4.

DETR treats the object detection task as an intuitive set prediction problem and discards some traditional hand-craft components such as hand-designed anchor sets and non-maximal suppression (NMS). As shown in Fig. 4, DETR uses CNN Backbone to learn the 2D features of the input image. Then feature maps are unfolded into sequences and fed to the Transformer encoder module (where there is still positional encoding). The output of the Transformer Decoder module is then obtained under the constraint of object queries. Finally, the class and bounding box regression parameters are obtained after a feedforward network.

Based on the idea of sequential prediction, DETR regards the prediction of the network as a fixed sequence $\tilde{y}$ of length N, $\tilde{y} = \tilde{y}_i, i \in (1, N)$, (where the value of N is fixed and much larger than the number of Ground Truth in the image), $\tilde{y}_i = \left(\tilde{c}_i, \tilde{b}_i\right)$. Meanwhile, the Ground Truth is considered as a sequence $y : y_i = \left(c_i, b_i\right)$ (the length must be less than N, so the sequence is filled with ϕ (for no object), which can be interpreted as the category of background, to make its length equal to N), where c_i denotes the true category to which the object belongs, and b_i denotes a quaternion (containing the center point coordinates and the width and height of the object box, and both are relative to the scale coordinates of the image).

So the prediction task can be viewed as a bipartite matching problem between y and $\tilde{y}$, with the Hungarian algorithm as the solution method, defining the strategy for minimum matching as follows:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathbb{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right), \tag{11}$$

where $\tilde{\sigma}$ denotes the matching strategy when finding the minimum loss, for $\mathbb{L}$ while considering the similarity prediction between Ground Truth boxes. For $\sigma(i), c_i$ the predicted category confidence is $\tilde{P}_{\sigma(i)}\left(c_i\right)$ and the bounding box prediction is $\tilde{b}_{\sigma(i)}$, for non-empty matches, define $\mathbb{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right)$ as: $-\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}\left(c_i\right) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathbb{L}_{\text{box}}\left(b_i, \hat{b}_{\sigma(i)}\right)$.

In this way, the overall loss is obtained as

$$\mathbb{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}\left(c_i\right) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathbb{L}_{\text{box}}\left(b_i, \hat{b}_{\hat{\sigma}}(i)\right)\right], \tag{12}$$

Considering the bounding box scale, the L_1 loss and the IoU loss are linearly combined to obtain the $\mathbb{L}_{box}$ loss:

$$\mathbb{L}_{box} = \lambda_{\text{iou}}\, \mathbb{L}_{\text{iou}}\left(b_i, \hat{b}_{\sigma(i)}\right) + \lambda_{\text{L1}} \left\| b_i - \hat{b}_{\sigma(i)} \right\|_1, \tag{13}$$

Table 4
Comparison between Transformer Necks and representative CNNs on COCO2017 Val set. "Multi-Scale" refers to multi-scale inputs. AP denotes IoU threshold = .50:.05:.95. AP_{50} and AP_{75} denote IoU threshold = .50 and .75. In addition, AP_S, AP_M, AP_L denote different scales of objects. Small means area < 32^{32}. Medium means 32^{32} < area < 96^{96}. Large means area > 96^{96}.

Method	Backbone	Epochs	GFLOPs	#Params(M)	Multi-scale	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN + FPN (Ren et al., 2016)	ResNet50	109	180	42	–	26	42.0	62.1	45.5	26.6	45.4	53.4
DERR+ (Carion et al., 2020)	ResNet50	500	86	41	–	28	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5+ (Carion et al., 2020)		500	187	41	–	12	43.4	63.1	45.9	22.5	47.3	61.1
DERR (Carion et al., 2020)		50	86	41	–	12	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5 (Carion et al., 2020)		50	187	41	–	12	43.4	63.1	45.9	22.5	47.3	61.1
UP-DETR (Dai et al., 2021a)	ResNet50	150	86	41	–	28	40.5	60.8	42.6	19.0	44.4	60.0
UP-DETR+ (Dai et al., 2021a)		300	86	41	–	28	42.8	63.0	45.3	20.8	47.1	61.7
Deformable DETR (Zhu et al., 2021)	ResNet50	50	173	40	–	19	43.8	62.6	47.7	26.4	47.1	58.0
Two-stage Deformable DETR (Zhu et al., 2021)		50	173	40	–	19	46.2	65.2	50.0	28.8	49.2	61.7
Conditional DETR (Meng et al., 2021)	ResNet50	108	90	44	–	–	43.0	64.0	45.7	22.7	46.7	61.5
Conditional DETR-DC5 (Meng et al., 2021)		108	195	44	–	–	45.1	65.4	48.5	25.3	49.0	62.2
ACT-MTKD(L=16) (Zheng et al., 2021)	ResNet50	–	156	–	–	14	40.6	–	–	18.5	44.3	59.7
ACT-MTKD(L=32) (Zheng et al., 2021)		–	169	–	–	16	43.1	–	–	22.2	47.1	61.4
SMCA (Gao et al., 2021)	ResNet50	50	152	40	–	10	43.7	63.6	47.2	24.2	47.0	60.4
SMCA+ (Gao et al., 2021)		50	152	108	–	10	45.6	65.5	49.1	25.9	49.3	62.6
Efficient DETR (Yao et al., 2021)	ResNet50	36	159	32	–	–	44.2	62.2	48.0	28.4	47.5	56.6
Efficient DETR* (Yao et al., 2021)		36	210	35	–	–	45.1	65.4	48.5	25.3	49.0	62.2
TSP-FCOS (Sun et al., 2021)	ResNet50	36	189	51.5	–	15	43.1	62.3	47.0	26.6	46.8	55.9
TSP-RCNN (Sun et al., 2021)		36	188	64	–	11	43.8	63.3	48.3	28.6	46.9	55.7
TSP-RCNN+ (Sun et al., 2021)		96	188	64	–	11	45.0	64.5	49.6	29.7	47.7	58.0
YOLOS-S (Fang et al., 2021)	DeiT-S	150	200	30.7	–	7	36.1	56.4	37.1	15.3	38.5	56.1
YOLOS-S (Fang et al., 2021)		150	179	27.9	–	5	37.6	57.6	39.2	15.9	40.2	57.3
YOLOS-B (Fang et al., 2021)	DeiT-B	150	537	127	–	–	42.0	62.2	44.5	19.5	45.3	62.1
PnP-DETR-R50-DC5-α-0.33 (Wang et al., 2021b)	ResNet50	500	20.7(omit backbone)	–	–	–	42.7	62.8	45.1	22.4	46.2	60.0
PnP-DETR-R50-DC5-α-0.5 (Wang et al., 2021b)		500	32.9(omit backbone)	–	–	–	43.1	63.4	45.3	22.7	46.5	61.1
Dynamic DETR (Dai et al., 2021c)	ResNet50	40	–	–	–	–	47.2	65.9	51.1	28.6	49.3	59.1
Anchor DETR-C5 (Wang et al., 2021c)	ResNet50	50	–	–	–	–	42.1	63.1	44.9	22.3	46.2	60.0
Anchor DETR-DC5 (Wang et al., 2021c)		50	–	–	–	–	44.2	64.7	47.5	24.7	48.2	60.6
D^{2ETR} (Lin et al., 2022)	PVT2	50	82	35	–	–	43.2	62.9	46.2	22.0	48.5	62.4
Deformable D^{2ETR} (Lin et al., 2022)		50	93	40	–	–	50.0	67.9	54.1	31.7	53.4	66.7
Sparse DETR- $\rho = 10\%$ (Roh et al., 2022)	ResNet50	50	105	41	–	25.3	45.3	65.8	49.3	28.4	48.3	60.1
Sparse DETR- $\rho = 10\%$ (Roh et al., 2022)	Swin-T	50	113	41	–	21.2	48.2	69.2	52.3	29.8	51.2	64.5
DAB-DETR (Liu et al., 2022a)	ResNet50	50	202	44	–	–	44.5	65.1	47.7	25.3	48.2	62.3
DAB-DETR* (Liu et al., 2022a)		50	216	44	–	–	45.7	66.2	49.0	26.1	49.4	63.1
DN-DETR (Li et al., 2022)	ResNet50	50	94	44	–	–	44.1	64.4	46.7	22.9	48.0	63.4
DN-DETR-DC5 (Li et al., 2022)		50	202	44	–	–	46.3	66.4	49.7	26.7	50.0	64.3
DN-Deformable-DETR (Li et al., 2022)		50	195	48	–	–	48.6	67.4	52.7	31.0	52.0	63.7
DINO-4scale (Zhang et al., 2022a)	ResNet50	12	279	47	–	24	47.9	65.3	52.1	31.2	50.9	61.9
DINO-5scale (Zhang et al., 2022a)		12	860	47	–	10	48.3	65.8	52.4	32.2	51.3	62.2
DINO-4scale (Zhang et al., 2022a)		36	–	–	–	–	50.5	68.3	55.1	32.7	53.9	64.9
DINO-5scale (Zhang et al., 2022a)		36	–	–	–	–	51.0	69.0	55.6	34.1	53.6	65.6
SAM-DETR (Zhang et al., 2022b)	ResNet50	50	100	58	–	–	39.8	61.8	41.6	20.5	43.4	59.6
SAM-DETR-DC5 (Zhang et al., 2022b)		50	210	58	–	–	43.3	64.4	46.2	25.1	46.9	61.0
Pix2Seq (Chen et al., 2021)	ResNet50	50	–	37	–	–	43.0	61.0	45.6	25.1	46.9	59.4
Pix2Seq-DC5 (Chen et al., 2021)		50	–	38	–	–	43.2	61.0	46.1	26.6	47.0	58.6

Additionally, we have presented the attention visualization of the encoder and decoder (as shown in Figs. 5 and 6). This visualization aids in understanding how the model focuses on various parts of the input image and utilizes attention mechanisms for object detection. The encoder processes the input image, captures its spatial information, and creates a set of contextualized feature representations. Attention visualization in the encoder demonstrates how the model concentrates on specific regions of the image, emphasizing crucial areas that contribute to the comprehension of the objects present. The decoder uses the encoded features to generate final object detections, employing a series of self-attention and cross-attention mechanisms to iteratively refine the predicted object bounding boxes and class labels.

In summary, DETR, the first Transformer-based end-to-end object detector, exhibited performance comparable to state-of-the-art (SOTA) methods at the time. However, there are evident drawbacks in its application: slow convergence and low accuracy on small objects. Nonetheless, its end-to-end architecture possesses significant potential and has attracted numerous researchers to explore improvements.

3.2.2. UP- DETR

Since DERT faces great challenges in training and optimization, it requires a huge amount of training data and an extremely long training schedule, which leads to limitations in application on small datasets. Moreover, the existing pretext task cannot be directly applied to train the Transformer module of DETR, because DETR focuses mainly on spatial localization rather than image instance-based or cluster-based segmentation learning. To address the above issues, Dai et al. (2021a) proposed UP-DETR, a DETR-like model capable of unsupervised pre-training, whose structure is shown in Fig. 7.

Multiple query patches are randomly cropped from a given image and the Transformer for detection is pre-trained to predict the bounding boxes of these query patches in the given image. In the pre-training

Fig. 5. Encoder self-attention for a set of reference points. It demonstrates the attention distribution after the input image is processed through the Transformer encoder.

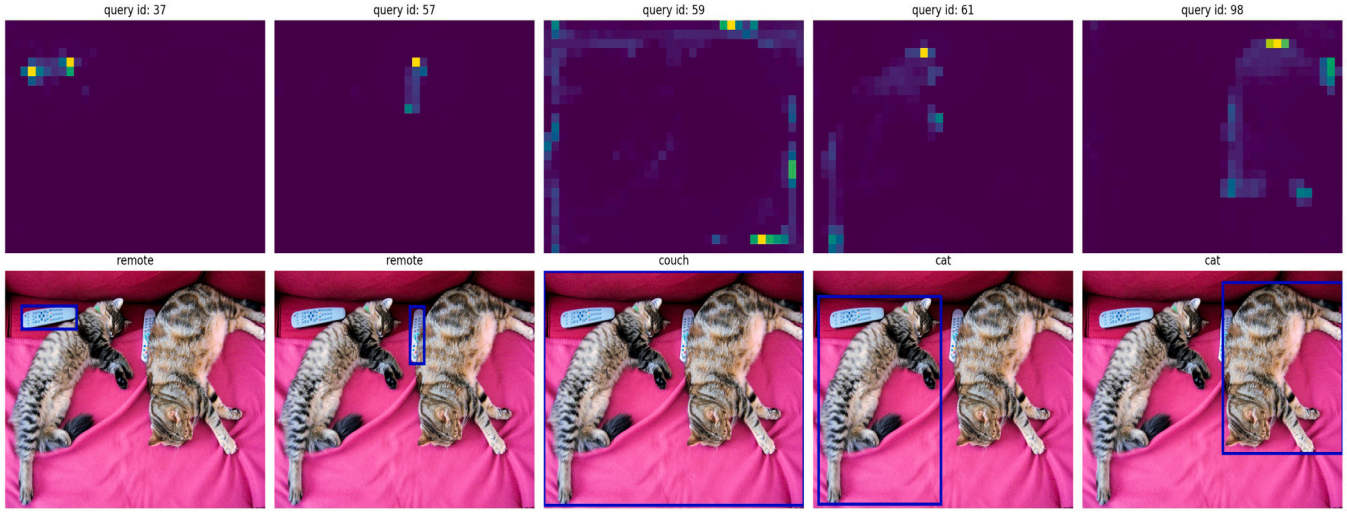
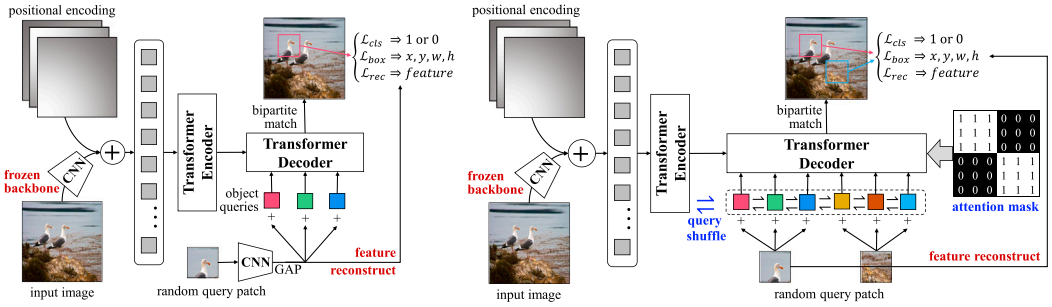


Fig. 6. Visualization of decoder attention for each predicted object in images from the COCO validation set, using the DETR-DC5 model. Attention scores are represented by distinct colors for different objects. The decoder primarily focuses on object extremities, such as legs and heads, highlighting the model's ability to capture fine-grained details. It is recommended to view this figure in color for better understanding.



(a) single-query patch (N=3, M=1) (b) multi-query patch (N=6, M=2)

Fig. 7. UP-DETR pre-training architecture by random query patch detection: (a) For one single-query patch, which is added to all object queries. (b) For the multi-query patch, which is added each query patch to N/M object queries with shuffle and attention mask.
Source: Image form Dai et al. (2021a).

process, the method addresses the following two key problems. (1) To trade-off the preference of classification and localization in the pre-text task, the backbone network is frozen and a patch feature reconstruction branch is proposed that is jointly optimized with patch detection. (2) For multi-query patch, UP-DETR is introduced in single-query patch and extended to multi-query patches with object query shuffle and attention mask.

In summary, UP-DETR proposes a new unsupervised pre-text task-random query patch detection to pre-train the Transformer. The results show that UP-DETR has significantly better performance than DETR in

object detection, panorama segmentation, and single detection, even on the PASCAL VOC dataset where the training data is insufficient.

3.2.3. YOLOS

Inspired by the pre-trained Transformer can fine-tune at the token level tasks (Rajpurkar et al., 2016; Sang and De Meulder, 2003), Fang et al. (2021) proposed YOLOS, a pure sequence-to-sequence transformer on the basis of DETR (Carion et al., 2020) and ViT (Dosovitskiy et al., 2021). It replaces the class token of the original ViT with the detection token and replaces the image classification loss with the bipartite matching loss of DETR in the training phase, which allows object detection by set prediction. YOLOS demonstrates the generality and transferability of the pre-trained Transformer from image classification to downstream object detection task, which is pre-trained in the classification task and then transfer to the detection task for fine-tuning. Experiments demonstrate that YOLOS-Base, pre-trained on only medium-sized ImageNet datasets can achieve 42.0 box AP.

3.2.4. Deformable DETR

Inspired by Deformable Convolution ((Dai et al., 2017), Zhu et al. (2021) proposed Deformable DETR. This method combines the advantages of sparse spatial sampling of deformable convolution with the relational modeling capability of Transformer. The Deformable Attention Module (DEM) is introduced to accelerate convergence and fuse multi-scale features to improve accuracy. Moreover, The authors introduce multi-scale feature from FPN (Lin et al., 2016), and then propose Multi-Scale Deformable Attention (MSDA) to replace the Transformer Attention Module for processing feature maps, as shown in Fig. 8 is shown. Let $\{\boldsymbol{x}^l\}_{l=1}^{L}$ be the input multi-scale feature map, where $x^l \in \mathbb{R}^{C\times H_l\times W_l}$. Let $\hat{\boldsymbol{p}}_q \in [0,1]^2$ be the normalized coordinates of the reference point of each query element q, and then compute Multi-Scale Deformable Attention as

$$\begin{aligned}&\mathrm{DeformAttn}\left(\boldsymbol{z}_q, \boldsymbol{p}_q, \boldsymbol{x}\right) = \\ &\sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k=1}^{K} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}\left(\boldsymbol{p}_q + \Delta \boldsymbol{p}_{mqk}\right)\right],\end{aligned} \tag{14}$$

where m is the index of attention head, l is the index of input feature level, and k is the index of sampling points. $\Delta \boldsymbol{p}_{mlqk}$ and A_{mlqk} denote respectively sampling offset and attention weight of the kth sampling point in the lth feature layer and the mth attention head. The scalar attention weights A_{mlqk} are normalized to $\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} = 1$. The normalized coordinates $(0,0)$ and $(1,1)$ of $\hat{\boldsymbol{p}}_q \in [0,1]^2$ denote the upper left and lower right corners of the image, respectively. The function $\phi_l\left(\hat{\boldsymbol{p}}_q\right)$ in Eq. (14) rescales the normalized coordinates $\boldsymbol{P}_q$ to the input feature map of the lth layer. The computational complexity of MSDA is $O\left(2N_qC^2 + \min\left(HWC^2, N_qKC^2\right)\right)$ compared to the original DETR, Deformable DETR requires less than one-tenth of training epochs to achieve better performance (especially on small object).

3.2.5. Conditional DETR

Meng et al. (2021) proposed Conditional DETR. They visualized experiments on the operation of DETR and concluded that cross-attention in DETR is highly dependent on content embedding to locate the four vertices and predict the bounding box. Thus, it increases the training difficulty. So they improved the cross-attention of DETR by concatenating the content query c_q and spatial query p_q, and the key by splicing the content key c_k and spatial key p_k. This inner product of query and key gives the following result:

$$\mathbf{c}_q^{\top}\mathbf{c}_k + \mathbf{p}_q^{\top}\mathbf{p}_k, \tag{15}$$

This separates the functions of content query and spatial query so that they focus on the weight of content and space respectively. As shown in Fig. 9, the improved Decoder layer consists of three main modules: 1) The self-attention layer, which is from the previous Decoder layer and is used to remove duplicate predictions as well as predict categories and bounding boxes; (2) The cross-attention layer, which can use embedding output of encoder to complete the embedding of the decoder; (3) The feed-forward networks layer (FFN).

The core of the conditional cross-attention mechanism is to learn a conditional spatial query from decoder embedding and reference points, which can explicitly find the boundary regions of the object, thus narrowing down the search object, helping to locate the object, and alleviating the problem of over-reliance on the quality of content embedding in DETR training. The problem of over-reliance on the quality of content embedding in DETR training is alleviated. These refinements improve the convergence speed of DETR by 8× faster and the box mAP on the COCO dataset by 1.8%.

3.2.6. Efficient DETR

Yao et al. analyzed the mechanisms of DETR and Deformable DETR and found that their common feature is a cascade structure stacked with six Decoders, which is used to iteratively update the object query. The reference point proposed by Deformable DETR visualizes the object query and solves the difficult problem that the object query is difficult to analyze directly. However, different initialization methods of reference points have a great impact on decoder performance. In order to investigate a more efficient way to initialize the object container, Yao et al. proposed Efficient DETR, a two-stage object detector that consists of dense prediction and sparse set prediction, and these two parts share the same detection head.

The model generates region proposals using dense detection before initializing the object container, and then uses the highest-scoring 4-dimensional proposal and its 256-dimensional encoder features as the initialization value of the object container, which results in better performance and fast convergence. The experimental results show that Efficient DETR combines the features of dense detection and ensemble detection, and can converge quickly while achieving high performance. The model achieves the SOTA performance at that time on the COCO dataset with only one encoder layer and three decoder layers, while the epoch is reduced by 14× less.

3.2.7. SMCA

To strengthen the relationship between the visual region of common interest for each object query and the bounding box to be predicted by the query, Gao et al. (2021) introduced spatial prior and multi-scale features, and proposed Spatially Modulated Co Attention (SMCA), which replaces the cross attention in the original Decoder while keeping the others unchanged.

The decoder of SMCA has multiple cross-attention heads, each of which estimates the object center and scale from a slightly different location, resulting in a series of different spatial weight maps. This weight map is used to spatially adjust the co-attention features, which improves the detection performance. Based on these improvements, SMCA can achieve 43.7 mAP in 50 epochs and 45.6 mAP in 108 epochs on the COCO dataset.

3.2.8. ACT

Due to the slow convergence of DETR, Zheng et al. (2021) proposed the Adaptive Clustering Transformer (ACT) to address the problem of high trial and error costs for improving DETR. ACT is a plug-and-play module that is fully compatible with Transformer and can be ported to DETR without any training. Its core design is first, to perform feature clustering adaptively for the attention redundancy (points with similar semantics and similar spatial locations produce similar attention maps) of encoder, select representative prototypes, and broadcast feature updates to their nearest neighboring points based on Euclidean distance. Second, an adaptive clustering algorithm is designed for the encoder note feature diversity problem (for different inputs, the feature distribution of each encoder layer is quite different), and a multi-round exact Euclidean location-sensitive hash (E2LSH) is chosen for this algorithm to adaptively determine the number of prototypes. Thanks

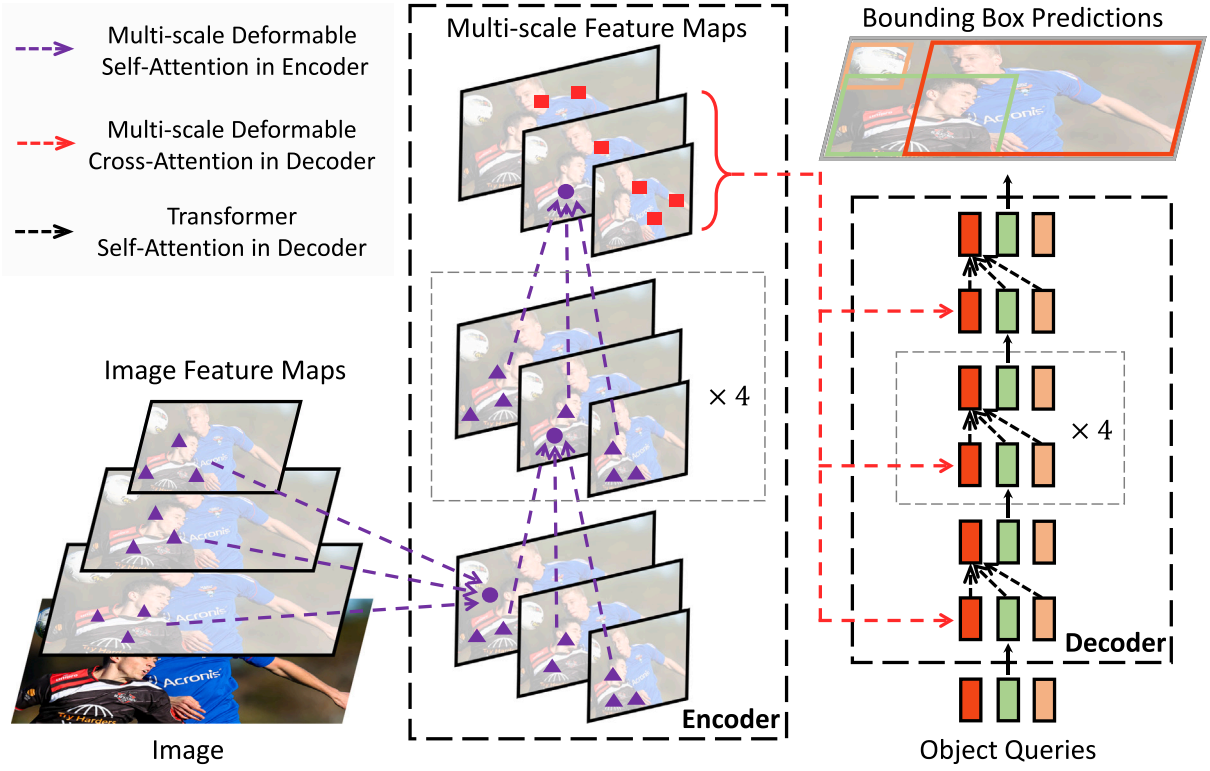


Fig. 8. The architecture of Deformable DETR. Its attention module focuses on only a small number of key sampling points around the reference point, and assigns a fixed and small number of keys to each object query, thus alleviating the problems of slow convergence and low feature resolution.
Source: Image from Zhu et al. (2021).

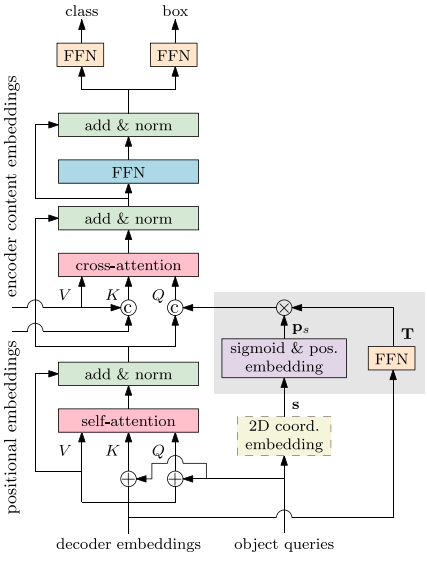


Fig. 9. A Decoder layer of Conditional DETR. The gray shaded box indicates that the Conditional spatial query is predicted from the learnable 2D coordinates s and the embedding output of the previous Decoder layer.
Source: Image from Meng et al. (2021).

to these improvements, ACT can reduce the FLOPS of DETR from 73.4 Gflops to 58.2 Gflops (excluding Backbone Resnet FLOPs) without additional training, while the loss of AP is only 0.7%. The AP loss can be further reduced to 0.2% by multitasking knowledge distillation. Given its excellent performance, exploring ACT training from scratch and fusion with multi-scale features is a worthy research direction in the future.

3.2.9. TSP

Sun et al. (2021) concluded after a lot of analysis that the cross-attention part of decoder and Hungarian loss of DETR are the main reasons for the slow convergence of DETR. So they proposed two improved models of DETR with only encoder, TSP-FCOS and TSP-RCNN corresponding to the One-Stage and Two-Stage object detection methods, respectively. Both models can be viewed as feature pyramid (Lin et al., 2016) based. The model uses a feature of interest (FoI) selection mechanism that helps encoder process multi-scale features. In addition, the model applies matching distillation to solve the instability of bipartite graph matching. Experiments show that TSP achieves better results with reduced training cost, using only 36-epoch to achieve the 500-epoch results of the original DETR training.

3.2.10. DINO

The Hungarian algorithm has been used in DETR (Carion et al., 2020) to match the output of the object by Decoder with Ground Truth. However, the discreteness of the Hungarian algorithm matching and the randomness of the model training cause the matching process to be dynamic and unstable, resulting the final slow convergence of DETR.

By deeply studying the iteration mechanism and optimization problems of the DETR model, Zhang et al. (2022a)proposed DINO (DETR with Improved deNoising anchor boxes) based on DN-DETR (Li et al., 2022), DAB-DETR (Liu et al., 2022a) and Deformable DETR (Zhu et al., 2021). The key design of DINO is that the training phase uses denoising training as a shortcut to learning the relative offset of anchor by first adding noise near the Ground Truth box, and then the Hungarian matching directly reconstructs the truth bounding box, thus improving the stability of matching. Secondly, the model also uses a query-based

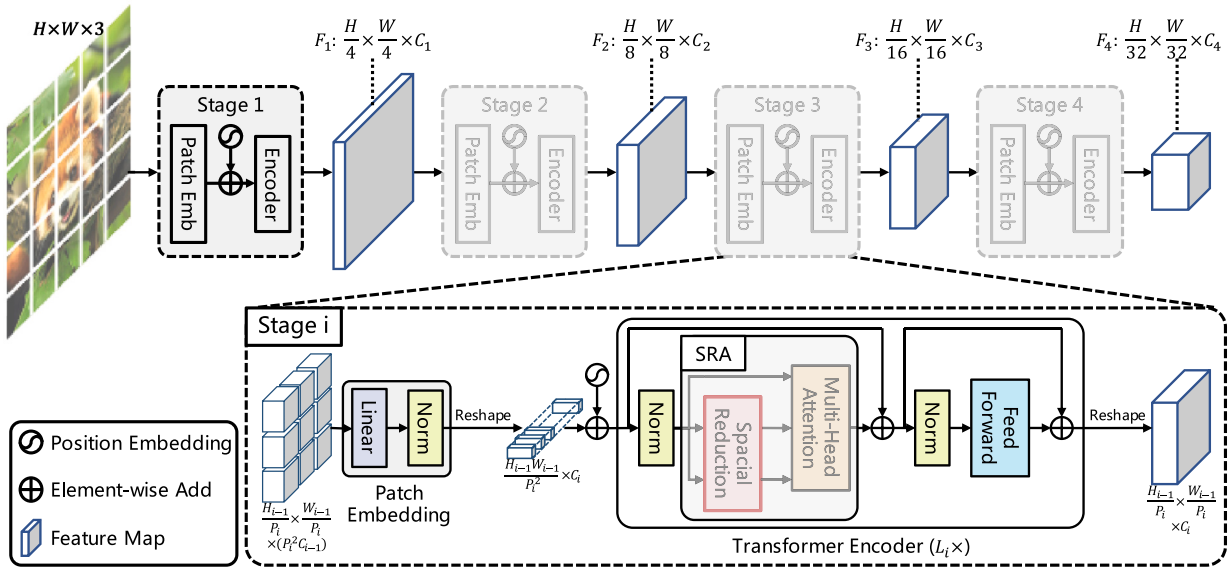


Fig. 10. The architecture of Pyramid Vision Transformer, the whole model is divided into 4 stages to generate feature maps at different scales. Each stage consists of a patch embedding layer, L_i-layer, and reshape operation.
Source: Image form Wang et al. (2021a).

dynamic anchor formulation to initialize the query and correct the parameters of adjacent earlier layers with the gradients of later layers. DINO breaks the dominance of classical architecture detector (SwinV2-G (Liu et al., 2021a), Florence (Yuan et al., 2021), DyHead (Dai et al., 2021b), etc.). DINO-Res50, which combines multi-scale features, achieves 48.3AP and 51.0AP on the COCO2017 dataset with 12-epoch and 36-epoch training schemes, respectively. Moreover, DINO-Swin-L even achieves the highest performance 63.3AP after training on a larger dataset.

3.3. Transformer backbone

Other efforts such as ViT (Dosovitskiy et al., 2021) have used Transformer in the image classification and achieved comparable results. However, there are some limitations in other complex CV tasks. These challenges of transferring the high performance of Transformer in NLP to the CV can be explained by the differences between the two domains.

1. The object entities in CV tasks often have dramatic scale variation.
2. Compared to text, the matrix nature of images makes it contain at least hundreds of pixels for an image that can express information. Especially the very long sequence unfolded by high-resolution images is difficult for Transformer to model.
3. Many CV tasks such as semantic segmentation require pixel-level dense prediction, and the computational complexity of the self-attention mechanism in ViT increases quadratically with image size, which leads to unacceptable computational overhead.
4. In the existing Transformer-based models, tokens are fixed in scale and not improved in design for CV tasks.

To address the above challenges, many Transformer-based backbones have been proposed for CV tasks and combined with methods such as multi-scale to compensate for the shortcomings that ViT can only detect at low resolution and so on. These methods can replace the backbone of mainstream object detection models, and in the benchmark Table 5, we list the performance of Mask R-CNN (He et al., 2017) and RetinaNet (Ross and Dollár, 2017) comparison after replacing the backbone and review the classical models in this subsection.

3.3.1. PVT&PVTv2

The feature maps output by ViT (Dosovitskiy et al., 2021) are difficult to apply to dense prediction due to their single scale and low resolution. Wang et al. (2021a) proposed the Pyramid Vision Transformer (PVT) by incorporating the multi-scale feature into Transformer. PVT can be used as a Backbone for various dense detection tasks, especially it can replace the CNN backbone of DETR-like models or be combined into a pure Transformer model without manual components such as NMS.

Benefiting from the progressive shrinking pyramid structure in the PVT, the Transformer sequence length decreases as the network gets deeper. Meanwhile, in order to further reduce the computation of fine-grained segmentation of images, they propose spatial-reduction attention (SRA) to reduce the computation of learning high-resolution feature maps (As shown in Fig. 10).

Compared with the CNN method based on feature pyramid structure, PVT not only generates multi-scale feature maps to detect objects of different sizes but also fuses global information through self-attention mechanism. The PVTv2 (Wang et al., 2022) proposed by the same team subsequently improves the PVT by adding a linear complexity attention layer, overlapping patch embedding, and convolutional feed-forward network to improve the performance of the PVT as backbone. On the COCO dataset, both achieved competitive results at that time.

3.3.2. Swin transformer

Liu et al. (2021b) proposed Swin Transformer, which creatively uses a hierarchical design to make the Transformer available as a backbone for most CV tasks, rather than just a detection head. As shown in Fig. 11, It is easy to see that, unlike other Transformer models, Swin Transformer builds a feature map with hierarchical representation, similar to the feature pyramid structure in CNN. As the network level deepens, the receptive expands, enabling the extraction of multi-scale features of the image. Secondly, Swin Transformer divides the feature map with multiple windows, and each non-overlapping window performs local multi-head attention calculation without correspondence between windows, which makes the computation greatly reduced and linear with the image size, as shown in the Eq. (16). In contrast, ViT produces a single low-resolution image and calculates global attention,

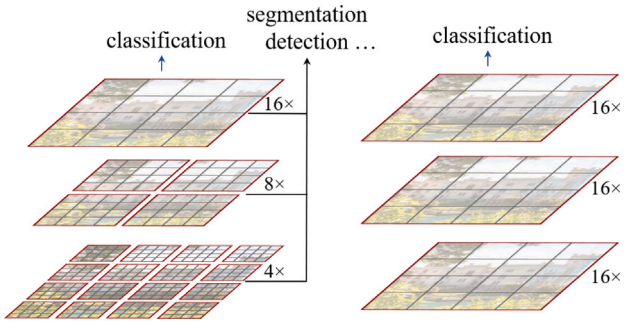


Fig. 11. Compare Swin Transformer (left) with ViT (right).
Source: Image form Liu et al. (2021b).

so the computational complexity and image size are quadratically related, as shown in Eq. (17)

$$\Omega(\mathrm{W-MSA}) = 4hwC^2 + 2M^2hwC, \tag{16}$$

$$\Omega(\mathrm{MSA}) = 4hwC^2 + 2(hw)^2C, \tag{17}$$

where M is a fixed window size (set to 7 by default), computing global attention for ViT is unacceptable for large image sizes HW, while window-based multi-head self-attention (W-MSA) is scalable.

The Pipeline of the Swin Transformer is shown in Fig. 12(a). The input image is spreading into a sequence after Patch Partition and Linear Embedding layers, and then input into 4 stages. The Swin Transformer block in each stage replaces the standard multi-head self-attention (MSA) module in the Transformer module with window-based self-attention (W-MSA) or a shift window-based module (SW-MSA), which introduces a relative position bias in the computation of attention to account for the geometric relationships in the self-attention computation, as shown in Eq. (18). This parameter accounts for the relative spatial configuration of the visual elements and is shown to be critical in various visual tasks, especially for intensive recognition tasks such as object detection and semantic segmentation.

$$\mathrm{Attention}(Q, K, V) = \mathrm{SoftMax}\left(QK^T/\sqrt{d} + B\right)V, \tag{18}$$

Although W-MSA reduces the computation greatly, W-MSA loses the ability to model the relationship between different windows, and the lack of information exchange between non-overlapping windows affects the representation of the model. So they introduced SW-MSA, which shifts the windows in the Swin Transformer Block of the next layer to introduce correspondence of the previous layer. This operation greatly increases the actual receptive field, as shown in Fig. 13. In this way, the multi-head attention is computed inside the new window to include the boundary of the original window and achieve the modeling of the relationship between windows.

But this approach causes the number of windows to change, and the window size is not uniform. An easy way to solve this problem is padding the small window, but it will increase the computation. So they proposed cyclic-shifting, a more efficient method of batch computation. This method cyclically shifts and merges small windows, so that a window may contain content from different windows, and therefore the masked MSA mechanism is used to restrict the self-attention computation to each sub-window, as shown in Fig. 14.

Swin Transformer has achieved SOTA performance on classification, detection, and segmentation tasks. Its biggest contribution is to propose a backbone that can be widely used in CV. And most of the hyperparameters commonly found in CNNs can be manually tuned in Swin Transformer, such as the number of network blocks and the size of input images. This method combines the advantages of both Transformer and CNN, fully considers the size invariance of CNN and the relationship between receptive field and number of layers, and solves the problem of slow application of Transformer in CV.

3.3.3. Swin TransformerV2

After Swin Transformer, Liu et al. (2021a) proposed Swin TransformerV2 to address the problems of expansion of CV models and training with high-resolution images, as well as the excessive GPU memory consumption for large models. Swin Transformer is optimized to scale up to 3 billion parameters and can be trained with images up to 1536 × 1536 resolutions. The improved method is shown in Fig. 15.

Post normalization technique: They found that when scaling up the model, the activation values in the deep layer increase dramatically. In fact, in the pre-normalized configuration (Layer Norm layer before the Attention layer), the output activation values of each residual block are directly merged back to the main branch, and the amplitude of the main branch becomes in the deeper layers. The huge amplitude differences between different layers may cause training instability problems. Therefore, they propose a post normalization technique, in which the output of each residual block is normalized before it is merged back into the main branch, and the amplitude of the main branch does not accumulate as the number of layers deepens.

Scaled cosine attention: In the original self-attention computation, the similarity terms of pixel pairs are computed as dot products of queries and keys vectors. However, when using this approach for large visual models, the learned attention graph for some blocks and attention heads is often dominated by several pixel pairs, especially in post-normalization configurations. To alleviate this problem, the authors propose a scaled cosine attention (Scaled cosine attention) method, which computes the number of attention pairs for a pixel pair i and j by a scaled cosine function:

$$\mathrm{Sim}\left(\mathbf{q}_i, \mathbf{k}_j\right) = \cos\left(\mathbf{q}_i, \mathbf{k}_j\right)/\tau + B_{ij}, \tag{19}$$

where B_{ij} is the relative position bias between pixels i and j; τ is a learnable scalar that cannot be shared across heads and layers. The τ is set to be greater than 0.01. The cosine function is naturally normalized so that it can have milder attention values, which improves the stability of large visual models and makes the model capacity easier to be scaled up.

Log-spaced continuous position bias: They found that the original relative position encoding method was weak for scale generalization of the model, and proposed log-spaced continuous position bias so that the relative position bias can be transferred smoothly across windows at different resolutions, effectively transferring models pre-trained in low-resolution images and windows to their higher resolution counterparts:

$$\begin{aligned}\widehat{\Delta x} &= \mathrm{sign}(x)\cdot\log(1+|\Delta x|),\\ \widehat{\Delta y} &= \mathrm{sign}(y)\cdot\log(1+|\Delta y|),\end{aligned} \tag{20}$$

where δx, δy and $\widehat{\Delta x}$, $\widehat{\Delta y}$ are the coordinates of linear scale and logarithmic space, respectively. The optimized resulting architecture was named Swin TransformerV2, and the model achieved a box/mask mAP of 63.1/54 in the COCO2017 dataset.

3.3.4. Other representative methods

In addition to the conventional approaches, our benchmark extends to include comparisons with several cutting-edge techniques. ViL, introduced by Zhang et al. (2021), realizes a multi-scale configuration through the sequential stacking of numerous ViT stages. Furthermore, it enhances the attention mechanism, thus elevating both efficiency and classification performance. The Focal Transformer introduces the novel Focal Self-Attention mechanism. This integrates both granular local and coarse global interactions, thereby ensuring the effective capturing of both proximal and distal visual dependencies. Twins (Chu et al., 2021) propose two highly efficient vision transformer architectures, Twins-PCPVT and Twins-SVT, both leveraging a restructured spatial attention mechanism. This methodology incorporates both locally-grouped self-attention and global sub-sampled attention, capturing both fine-grained proximal and coarse-grained distal visual information. Dong et al. (2022) introduced the CSWin Transformer, a robust Transformer-based

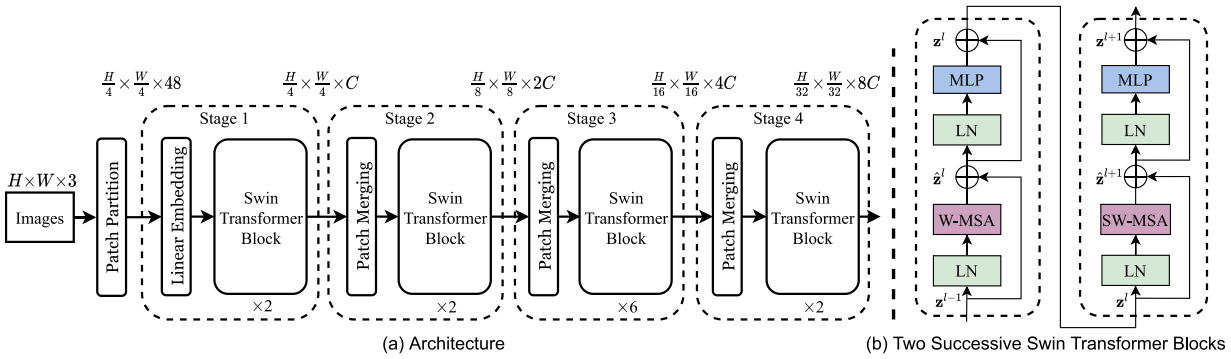


Fig. 12. (a) Swin Transformer (Swin-T) (b) Swin Transformer Block.
Source: Image from Liu et al. (2021b).

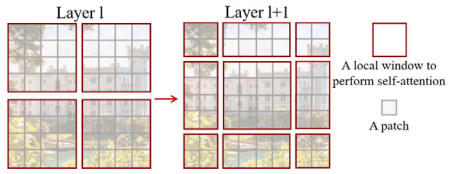


Fig. 13. The shift window approach can calculates the self-attention across the window boundary of the previous layer.
Source: Image from Liu et al. (2021b).

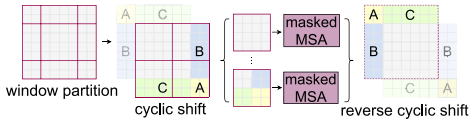


Fig. 14. An illustration of circular shift.
Source: Image from Liu et al. (2021b).

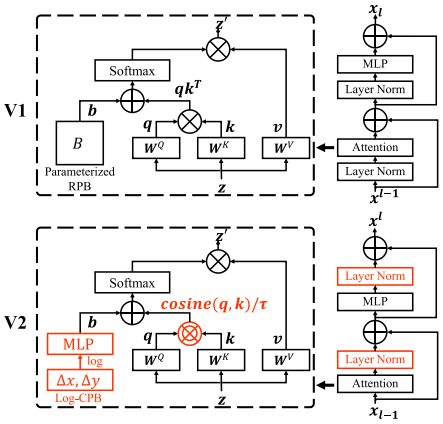


Fig. 15. Comparison of the attention modules of Swin TransformerV1 and V2.
Source: Image from Liu et al. (2021a).

backbone for vision tasks. It integrates the Cross-Shaped Window self-attention mechanism, varies stripe widths based on network depth, and introduces a novel Locally-enhanced Positional Encoding (LePE) scheme to handle local positional information optimally, resulting in competitive performance across standard vision tasks.

3.4. Analysis and discussion for detectors

This section provides a succinct review of conventional Transformer-based object detectors, offering a detailed performance comparison in Tables 4 and 5. Each method was evaluated using the NVIDIA A100 GPU and adhered to the DETR training protocol. The AdamW optimizer (Loshchilov and Hutter, 2017) was uniformly employed across all methods, with the initial learning rate for the transformer set to 10^{-4}, the backbone's to 10^{-5}, and weight decay at 10^{-4}. The transformer weights were initialized with Xavier init (Glorot and Bengio, 2010), while the backbone leveraged the ImageNet-pretrained ResNet model from torchvision, with frozen batch normalization layers.

For Transformer Neck-based models, they treat object detection as a straightforward set prediction, removing manual components (such as anchor set and NMS) that cannot be optimized, thus enabling end-to-end detection. Starting from the original DETR with slow convergence and poor detection of small objects, subsequent researchers have proposed optimization strategies from different perspectives.

1. To address the problem of slow convergence, researchers often start by improving the attention mechanism. Deformable DETR (Zhu et al., 2021) accelerates convergence 12× faster with the Deformable Attention Module. Conditional DETR improves the cross-attention of DETR and gets 8× faster convergence. Meanwhile, the box mAP on the COCO dataset is improved by 1.8%. Unlike the above methods, ACT (Zheng et al., 2021) proposes a plug-and-play module for adaptive clustering, which reduces the GFLOPS of DETR by 15.2 without additional training, while the AP loss is only 0.7%. Sparse DETR achieves higher performance and the same detection speed (FPS) as Faster R-CNN by improving and reducing the GFLOPs by 75.

2. For the problem of poor detection of small objects, multi-scale feature is currently the main focus. Methods such as SMCA (Gao et al., 2021) (as shown in Table 4) introduce multi-scale feature with different operations and significantly improve the accuracy of the detector. Moreover, DINO (Zhang et al., 2022a) reaches 63.3 AP over all classical object detection methods.

Presently, most Transformer Backbones are primarily active in image classification, with only a few researchers transitioning them to traditional object detectors for dense prediction. These have then achieved state-of-the-art (SOTA) performance. Compared to CNN-based Backbones, Transformer-based Backbones can integrate global contextual information while outputting multi-scale feature maps, thereby enhancing feature extraction. Although Transformers have challenged CNN's dominance in object detection, recent advancements such as FAIR's redesign of ConvNet (Liu et al., 2022b), which draws from the strengths of the Transformer structure, underscore the continued potential of CNNs. In the future, CNN and visual Transformer are expected to continue improving by leveraging each other's strengths.

4. Discussion

Although the Transformer model has made great progress (as shown in Table 6) and has shown excellent performance (Table 4, Table 5), they still face some challenges, as well as limitations in practical

Table 5
The prediction results of RetinaNet and Mask R-CNN with Transformer as Backbone on COCO 2017 Val Set. Where 3 × schedule denotes 36-epoch, MS denotes multi-scale input (MS), and the numbers before and after "/" denote the parameters of RetinaNet and Mask R-CNN, respectively.

Backbone	#Params	FlOPs	RetinaNet 3×schedule + MS						Mask R-CNN 3×schedule + MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 (He et al., 2015)	38/44	239/260	39	58.4	41.8	22.4	42.8	51.6	41	61.7	44.9	37.1	58.4	40.1
PVTv1-S (Wang et al., 2021a)	34/44	226/245	42.2	62.7	45.0	26.2	45.2	57.2	43.0	65.3	46.9	39.9	62.5	42.8
ViL-S (Zhang et al., 2021)	36/45	252/174	42.9	63.8	45.6	27.8	46.4	56.3	43.4	64.9	47.0	39.6	62.1	42.4
Swin-T (Liu et al., 2021b)	39/48	245/264	45.0	65.9	48.4	29.7	48.9	58.1	46.0	68.1	50.3	41.6	65.1	44.9
PVTv2-B2-Li (Liu et al., 2021b)	32/42	-/-	–	–	–	–	–	–	46.8	68.7	51..4	42.3	65.7	45.4
Focal-T (Yang et al., 2021)	39/49	265/291	45.5	66.3	48.8	31.2	49.2	58.7	47.2	69.4	51.9	42.7	66.5	45.9
TwinsP-S (Chu et al., 2021)	34/44	-/245	45.2	66.5	48.6	30.0	48.8	58.9	46.8	69.3	51.8	42.6	66.3	46.0
Twins-S (Chu et al., 2021)	34/55	-/228	45.6	67.1	48.6	29.8	49.3	60.0	46.8	69.2	51.2	42.6	66.3	45.8
CSwin-T (Dong et al., 2022)	-/42	-/279	–	–	–	–	–	–	49.0	70.7	53.7	43.6	67.9	46.6
PVTv2-B2 (Wang et al., 2022)	35/45	-/-	–	–	–	–	–	–	47.8	69.7	52.6	43.1	66.8	46.7
ResNet101 (He et al., 2015)	57/63	315/336	40.9	60.1	44.0	23.7	45.0	53.8	42.8	63.2	47.1	38.5	60.1	41.3
ResNeXt101-32 × 4d (He et al., 2015)	56/63	319/340	41.4	61 .0	44.3	23.9	45.5	53.7	44.0	64.4	48.0	39.2	61.4	41.9
PVTv1-M (Wang et al., 2021a)	54/64	283/302	43.2	63.8	46.1	27.3	46.3	58.9	44.2	66.0	48.2	40.5	63.1	43.5
ViL-M (Zhang et al., 2021)	51/60	339/261	43.7	64.6	46.4	27.9	47.1	56.9	44.6	66.3	48.5	40.7	63.8	43.7
TwinsP-B (Chu et al., 2021)	54/64	-/302	46.4	67.7	49.8	31.3	50.2	61.4	47.9	70.1	52.5	43.2	67.2	46.3
Twins-B (Chu et al., 2021)	67/76	-/340	46.9	68.0	50.2	31.7	50.3	61.8	48.0	69.5	52.7	43.0	66.8	46.6
Swin-Scite (Liu et al., 2021b)	60/69	335/354	46.4	67.0	50.1	31.0	50.1	60.3	48.5	70.2	53.5	43.3	67.3	46.6
Focal-S (Yang et al., 2021)	62/71	367/401	47.3	67.8	51.0	31.6	50.9	61.1	48.8	70.5	53.6	43.8	67.7	47.2
CSwin-S (Dong et al., 2022)	-/54	-/342	–	–	–	–	–	–	50.0	71.3	54.7	44.5	68.4	47.7
ResNeXt101-64 × 4d (He et al., 2015)	96/102	473/493	41.8	61.5	44.4	25.2	45.4	54.6	44.4	64.9	48.8	39.7	61.9	42.6
PVTv1-Large (Wang et al., 2021a)	71/81	345/364	43.4	63.6	46.1	26.1	46.0	59.5	44.5	66.0	48.3	40.7	63.4	43.7
ViL-Base (Zhang et al., 2021)	67/76	443/365	44.7	65.5	47.6	29.9	48.0	58.1	45.7	67.2	49.9	41.3	64.4	44.5
Swin-Base (Liu et al., 2021b)	98/107	477/496	45.8	66.4	49.1	29.9	49.4	60.3	48.5	69.8	53.2	43.4	66.8	46.9
Focal-Base (Yang et al., 2021)	101/110	514/533	46.9	67.8	50.3	31.9	50.3	61.5	49.0	70.1	53.6	43.7	67.6	47.0
CSWin-B (Dong et al., 2022)	-/97	-/526	–	–	–	–	–	–	50.8	72.1	55.8	44.9	69.1	48.3

applications. This section will summarize the innovative improvements of the current method, analyze the problems encountered by the Transformer detector, and give an outlook on the future development prospects.

4.1. Challenges

High computational overhead. Typical properties of CNNs include inductive bias, which is expressed as translation invariance, weight sharing, and sparse connectivity (Dosovitskiy et al., 2021). These properties grant CNNs a robust local feature extraction capability and enable them to achieve high performance through the simple sliding matching of convolutional kernels. As a result, compared to Transformers, CNNs often exhibit competitive performance with lower computational overhead. However, current CNN architectures possess less potential than Transformers due to their weaker extraction of global features and contextual information. The self-attention mechanism in Transformers can also emulate convolutional layers, requiring only a sufficient number of heads to focus on each pixel within the convolutional receptive field and employing relative positional encoding to ensure translation invariance (Cordonnier et al., 2020). This full-attention operation can effectively integrate local and global attention while dynamically generating attention weights based on feature relationships. Nevertheless, Transformers face certain limitations in practical applications. One of the main challenges stems from their high computational complexity.

The expensive computational overhead restricts the application of Transformer-based detectors on mobile computing platforms. At present, most mobile detection platforms primarily rely on one-stage detectors (Zhao et al., 2019), while the trend for Transformer detectors leans towards offline high-precision detection. Additionally, Transformers require large amounts of data, and common solutions include data augmentation, self-supervised, or semi-supervised learning approaches (He et al., 2021). Compared to state-of-the-art CNN-based approaches, their deployment on mobile platforms is constrained by higher computational complexity.

The impact of computational overhead on deploying Transformer-based object detection models in practical scenarios is influenced by factors such as the number of parameters, running time (FPS), and floating-point operations (FLOPs). However, these metrics' influence varies depending on the application context and hardware environment. For example, in situations like autonomous driving or robotic navigation, FPS is a critical factor, as algorithms must process video streams at high frame rates to respond quickly to external changes. In the case of mobile devices and embedded systems, the number of parameters and FLOPs are more influential due to energy and memory constraints. Consequently, deploying algorithms on mobile platforms necessitates balancing performance, energy consumption, and memory usage. In cloud computing and high-performance hardware settings, computational overhead is not the most critical factor since computational resources are relatively abundant. In these scenarios, model performance and accuracy are paramount.

According to the data in Table 4, Table 5, modern Transformer-based models have outperformed classical two-stage object detection algorithms (e.g., Faster R-CNN) in terms of FPS and achieved improved accuracy, rendering them viable for practical applications. To ensure efficient deployment and application in real-world engineering scenarios, researchers typically optimize object detection algorithms for specific contexts, minimizing computational overhead and improving real-time performance and energy efficiency. This optimization may involve techniques such as model compression, knowledge distillation, and network architecture design.

Insufficient understanding of visual Transformer. Compared to the well-established research and applications of CNNs, our current understanding of the underlying mechanisms behind visual Transformers is still limited. The Transformer architecture was originally designed for sequence processing tasks (Vaswani et al., 2017). Although Transformers have demonstrated strong performance when applied to computer vision tasks, there is relatively little explanation regarding their specific roles and functions in this context. Consequently, gaining a deeper understanding of the principles behind visual Transformers is crucial to facilitate more fundamental optimization improvements and enhance the model's interpretability. This deeper understanding could potentially involve investigating the attention mechanisms, hierarchical feature representation, and the interaction between different layers

Table 6
Summary of the advantages and limitations of Transformer-based object detection models.

Type	Method	Highlights	Limitations
Transformer Neck	DETR (Carion et al., 2020)	(1) Proposed Transformer-based end-to-end object detection framework, (2) Removed hand-designed anchor set and non-maximal suppression (NMS)	(1) Requires massive dataset training, (2) Convergence is very slow, (3) Poor performance for small objects.
	SMCA (Gao et al., 2021)	(1) Combining a learnable co-attention map and a manual space prior speeds up the convergence of DETR, (2) Incorporating a scale selection network in decoder.	(1) Good performance for large objects and poor performance for small objects, (2) High computational overhead.
	Deformable DETR (Zhu et al., 2021)	(1) Proposed deformable attention mechanism, which pays more attention to local information and improves convergence speed; (2) Combined with multi-scale feature, (3) Proposed reference point visualization object query, (4) Two-stage Deformable DETR is also proposed.	(1) Low accuracy for large objects, (2) Deformable attention brings unordered memory access, (3) High computational overhead.
	Efficient DETR (Yao et al., 2021)	(1) They found that different object container initialization methods have a great impact on decoder; (2) They also proposed an efficient way of initializing object containers using the characteristics of dense detection and sparse detection.	(1) Poor performance for small objects, (2) High computational overhead.
	DINO (Zhang et al., 2022a)	(1) Propose a contrast denoising training method, (2) Combine class DETR and two-stage model and propose a mixed query selection method to better initialize object query, (3) Look Forward Twice: Introducing proximity layer information to update parameters and improve the detection of small objects.	(1) High computational overhead at high scales, (2) Diminishing marginal benefit from stacking too many scales.
	YOLOS (Fang et al., 2021)	(1) Replace [cls] token with [det] token and image classification loss with bipartite matching loss, (2)Propose a pre-trained Transformer object detection paradigm.	(1) Low detection accuracy, (2) High computational overhead.
	UP-DETR (Dai et al., 2021a)	(1) Propose a new unsupervised pre-text task to perform unsupervised pre-training on Transformer, (2) Propose a patch detection reconstruction branch that is jointly optimized with patch detection.	(1) Slow convergence, (2) Poor performance for small objects.
Transformer Backbone	FPT (Zhang et al., 2020)	(1) Propose a feature interaction method across space and scale, (2) High compatibility.	(1) Low detection accuracy, (2) High computational overhead.
	PVT (Wang et al., 2021a)	(1) It can output multi-scale high-resolution feature maps; (2) The proposed spatial reduction attention module makes PVT successfully applied to dense prediction.	(1) High computational overhead for high-resolution images; (2) Simple image division loses the connection information between different patches.
	Swin Transformer (Liu et al., 2021b)	(1) Hierarchical representation, (2) Introduced communication between windows by computing attention within shifted windows and reduced the computational complexity to be linear with the image size.	(1) Excessive GPU memory consumption at higher image resolutions, (2) Difficult to retrain on small datasets, (3) Difficult to transform pre-trained models at low resolutions to higher resolutions.

within the visual Transformer models. By exploring these aspects, we can potentially uncover novel optimization strategies and improve the model's overall performance in various computer vision tasks.

The inefficient image-sequence information transformation. Unlike images, human-created languages have a high semantic density. Each word in a sentence can be treated as high-dimensional semantic information embedded in a low-dimensional vector representation. However images, as a natural signal with high spatial redundancy, have a low information density per pixel. For example, He et al. (2021) performed random high-scale masking of images, and then reconstructed the images well with Decoder, demonstrating that much higher semantic features than pixel information density can be captured in the images. But the current way of representing image information in sequences using Transformer is not efficient enough, which can bring about accuracy degradation as well as high computational overhead. Establishing efficient transformations of image sequences can help unlock the potential of the Transformer for CV tasks.

4.2. Future development outlook

Visual Transformer has made great progress in recent years, especially in object detection, and the performance has surpassed SOTA CNN-based model on the COCO dataset. However, Transformer is not mature enough in practical application deployment. For example, the computational overhead is too large to be deployed on platforms with limited computer resources, and the real-time performance is not as good as the CNN-based one-stage approach.

Self-supervised learning. While self-supervised learning has made a great success in natural language processing, current object detection models, which are mainly supervised learning, require large amounts

of high-quality manually labeled data, which is usually too expensive. Therefore, It is natural to think of using self-supervised learning for visual tasks in order to pre-training models using a large amount of cheap data available on the Internet. For example, the MAE proposed by He et al. (2021) uses masked self-encoders for self-supervised learning, which are adequately pre-trained and then migrated to specific tasks for fine-tuning.

Lightweight Transformer. Since the performance of the current Transformer-based detector is powerful enough, the development of a lightweight Transformer architecture should be considered to broaden its applicability. Key considerations could include a reduction in computational demands, optimization for object detection, and intelligent query design to ensure high performance while minimizing computational overhead. This would enable deployment on mobile platforms with limited computational resources.

Multitasking. Within CNN-based methods, Mask R-CNN (He et al., 2017) successfully performs instance segmentation alongside object detection, yielding superior results. Could a Transformer detector also undertake multiple tasks simultaneously and derive benefits from this approach? For instance, the performance of the object detector could be enhanced by incorporating semantic segmentation. Semantic segmentation captures object boundaries, aiding object localization in detection, and segments the background to delineate the contextual information of the object, improving detection probability. Such an approach is especially useful as objects typically exist within specific contexts, such as cars appearing on roads.

5. Conclusion

For the past decade, CNN-based models have reigned supreme in the field of object detection. However, the Transformer has recently demonstrated superior performance and substantial potential in computer vision (CV), rendering Transformer-based models a burgeoning research topic within object detection. In this paper, we have conducted an extensive review of mainstream Transformer-based object detectors developed over the past three years. Our focus has mainly been on their concepts, innovative aspects, and detection accuracy. We have categorized these methods according to their model structure and established a benchmark based on the COCO2017 dataset. Furthermore, we have conducted a multi-perspective analysis and comparison of these methods, summarizing their innovations and enhancements. We have also provided a comprehensive analysis of their limitations and summarized the existing challenges that persist within the application of Transformer in object detection. This study aims to aid readers in deepening their understanding of Transformer object detectors, sparking research interest to unleash the potential of the Transformer model, and enhancing its practical applications.

CRediT authorship contribution statement

Yong Li: Writing – review & editing, Supervision, Project administration. **Naipeng Miao:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing- – original draft, Writing – review & editing, Visualization. **Liangdi Ma:** Writing – review & editing. **Feng Shuang:** Funding acquisition, Resources. **Xingwen Huang:** Writing – review & editing.

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

The data is public.

References

Arkin, Ershat, Yadikar, Nurbiya, Muhtar, Yusnur, Ubul, Kurban, 2021. A survey of object detection based on CNN and transformer. In: 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML). pp. 99–108. http://dx.doi.org/10.1109/PRML52754.2021.9520732.

Arkin, E., Yadikar, N., Xu, X., Aysa, A., Ubul, K., 2022. A survey: object detection methods from cnn to transformer. Multimedia Tools Appl. http://dx.doi.org/10.1007/s11042-022-13801-3.

Bai, Y., Mei, J., Yuille, A., Xie, C., 2021. Are transformers more robust than CNNs? http://dx.doi.org/10.48550/arXiv.2111.05464.

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. YOLOv4: Optimal speed and accuracy of object detection.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., 2020. Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 33, 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers.

Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1907–1915.

Chen, T., Saxena, S., Li, L., Fleet, D.J., Hinton, G., 2021. Pix2seq: A language modeling framework for object detection. arXiv:2109.10852[cs].

Chen, C., Seff, A., Kornhauser, A., Xiao, J., 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2722–2730.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C., 2021. Twins: Revisiting the design of spatial attention in vision transformers. arXiv:2104.13840[cs].

Cordonnier, J.-B., Loukas, A., Jaggi, M., 2020. On the relationship between self-attention and convolutional layers. arXiv:1911.03584[cs, stat].

Dai, Z., Cai, B., Lin, Y., Chen, J., 2021a. UP-DETR: Unsupervised pre-training for object detection with transformers. arXiv:2011.09094[cs].

Dai, X., Chen, Y., Xiao, B., Chen, D., Liu, M., Yuan, L., Zhang, L., 2021b. Dynamic head: Unifying object detection heads with attentions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7373–7382.

Dai, X., Chen, Y., Yang, J., Zhang, P., Yuan, L., Zhang, L., 2021c. Dynamic DETR: End-to-end object detection with dynamic attention. In: 2021 IEEE/CVF International Conference on Computer Vision. ICCV, IEEE, Montreal, QC, Canada, pp. 2968–2977. http://dx.doi.org/10.1109/ICCV48922.2021.00298.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y., 2017. Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 764–773.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dollar, P., Wojek, C., Schiele, B., Perona, P., 2012. Pedestrian detection: An evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 34 (4), 743–761, doi:10/bjsn5q.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B., 2022. CSWin transformer: A general vision transformer backbone with cross-shaped windows. arXiv:2107.00652[cs].

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv:2010.11929[cs].

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2007. The PASCAL visual object classes challenge 2007 (VOC2007) results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2012. The PASCAL visual object classes challenge 2012 (VOC2012) results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., Liu, W., 2021. You only look at one sequence: Rethinking transformer in vision through object detection. arXiv:2106.00666[cs].

Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1627–1645, doi: 10/fgv7fd.

Gao, P., Zheng, M., Wang, X., Dai, J., Li, H., 2021. Fast convergence of DETR with spatially modulated co-attention. arXiv:2101.07448[cs].

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. YOLOX: Exceeding YOLO series in 2021.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W., 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv:1811.12231[cs, q-bio, stat].

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. In: JMLR Workshop and Conference Proceedings, pp. 249–256.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., Tao, D., 2022. A survey on vision transformer. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. p. 1. http://dx.doi.org/10.1109/TPAMI.2022.3152247.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2021. Masked autoencoders are scalable vision learners.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. arXiv:1512.03385[cs].

Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M., 2021. Transformers in vision: A survey. arXiv:2101.01169.

Kobatake, H., Yoshinaga, Y., 1996. Detection of spicules on mammogram based on skeleton analysis. IEEE Trans. Med. Imaging 15 (3), 235–245. http://dx.doi.org/10.1109/42.500062.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2019. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L., 2022. DN-DETR: Accelerate DETR training by introducing query denoising. arXiv:2203.01305[cs].

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2016. Feature pyramid networks for object detection. arXiv:1612.03144.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: European Conference on Computer Vision. Springer, pp. 740–755.

Lin, J., Mao, X., Chen, Y., Xu, L., He, Y., Xue, H., 2022. D2ETR: Decoder-only DETR with computationally efficient cross-scale attention. arXiv:2203.00860[cs].

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. SSD: Single shot multibox detector. In: Computer Vision – ECCV 2016. pp. 21–37. http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., Guo, B., 2021a. Swin transformer V2: Scaling up capacity and resolution. arXiv:2111.09883[cs].

Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L., 2022a. DAB-DETR: Dynamic anchor boxes are better queries for DETR. arXiv:2201.12329[cs].

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021b. Swin transformer: Hierarchical vision transformer using shifted windows. arXiv:2103.14030[cs].

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022b. A ConvNet for the 2020s. arXiv:2201.03545[cs].

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. RoBERTa: A robustly optimized BERT pretraining approach. arXiv:1907.11692[cs].

Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., He, Z., 2021c. A survey of visual transformers. arXiv:2111.06091[cs].

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., Wang, J., 2021. Conditional DETR for fast training convergence. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3651–3660.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., 2018. Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. Openai Blog 1 (8), 9.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuAD: 100, 000+ questions for machine comprehension of text. arXiv:1606.05250[cs].

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, IEEE, Las Vegas, NV, USA, pp. 779–788, doi:10/gc7rk9.

Redmon, J., Farhadi, A. and, 2017. YOLO9000: Better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 6517–6525, DOI: 10/gffdbj.

Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards Real-Time Object Detection with region proposal networks. arXiv:1506.01497 [cs].

Roh, B., Shin, J., Shin, W., Kim, S., 2022. Sparse DETR: Efficient end-to-end object detection with learnable sparsity. arXiv:2111.14330[cs].

Ross, T.-Y., Dollár, G., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE Conference on Computer Visionv and Pattern Recognition. pp. 2980–2988.

Sang, E.F.T.K., De Meulder, F., 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv:cs/0306050.

Sun, Z., Cao, S., Yang, Y., Kitani, K.M., 2021. Rethinking transformer-based set prediction for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3611–3620.

Sung, K.-K., Poggio, T., 1998. Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal. Mach. Intell. 20 (1), 39–51. http://dx.doi.org/10.1109/34.655648/bnkgmt.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. 9.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, Vol. 30. Curran Associates, Inc.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2021a. Pyramid vision transforme: A versatile backbone for dense prediction without convolutions. arXiv:2102.12122[cs].

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L., 2022. PVTv2: Improved baselines with pyramid vision transformer. arXiv:2106.13797[cs].

Wang, T., Yuan, L., Chen, Y., Feng, J., Yan, S., 2021b. PnP-DETR: Towards efficient visual analysis with transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4661–4670.

Wang, Y., Zhang, X., Yang, T., Sun, J., 2021c. Anchor DETR: Query design for transformer-based object detection.

Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J., 2021. Focal self-attention for local-global interactions in vision transformers. arXiv:2107.00641[cs].

Yao, Z., Ai, J., Li, B., Zhang, C., 2021. Efficient DETR: Improving end-to-end object detector with dense prior. arXiv:2104.01318[cs].

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., Zhang, P., 2021. Florence: A new foundation model for computer vision. arXiv:2111.11432[cs].

Zhang, P., Dai, X., Yang, J., Xiao, B., Yuan, L., Zhang, L., Gao, J., 2021. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. arXiv preprint arXiv:2103.15358.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.-Y., 2022a. DINO: DETR with improved denoising anchor boxes for end-to-end object detection. arXiv:2203.03605[cs].

Zhang, G., Luo, Z., Yu, Y., Cui, K., Lu, S., 2022b. Accelerating DETR convergence via semantic-aligned matching. arXiv:2203.06883[cs].

Zhang, D., Zhang, H., Tang, J., Wang, M., Hua, X., Sun, Q., 2020. Feature pyramid transformer. In: European Conference on Computer Vision. Springer, pp. 323–339.

Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X., 2019. Object detection with deep learning: A review. IEEE Trans. Neural Netw. Learn. Syst. 30 (11), 3212–3232. http://dx.doi.org/10.1109/TNNLS.2018.2876865.

Zheng, M., Gao, P., Zhang, R., Li, K., Wang, X., Li, H., Dong, H., 2021. End-to-end object detection with adaptive clustering transformer. arXiv:2011.09315[cs].

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J., 2021. Deformable DETR: Deformable transformers for end-to-end object detection. arXiv:2010.04159[cs].