ISBN: 978-981-18-7950-0
2023 the 13th International Workshop on Computer Science and Engineering (WCSE 2023)
doi: 10.18178/wcse.2023.06.016

Fully Transformer Detector with Multiscale Encoder and Dynamic Decoder

Dengdi Sun [1,4], Zhen Hu[2], Bin Luo[2] and Zhuanlian Ding [3 +]

[1] School of Artificial Intelligence, Anhui University, Hefei, 230601, China.

[2] School of Computer Science and Technology, Anhui University, Hefei, 230601, China.

[3] School of Internet, Anhui University, Hefei, 230039, China.

[4] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, 230026, China.

Abstract. The recently proposed Detection Transformer (DETR) model applies the transformer encoder and decoder architecture to object detection and achieves comparable performance with CNN-based detection frameworks. However, DETR and other relevant variants usually use CNNs as backbone so that the output features of backbone are unfriendly to transformer encoders and decoders. Therefore, we propose a CNN-free end-to-end detector completely based on Transformer encoder and decoder. In addition, most detector based on transformer encoder and decoder problems lie in two aspects: slow convergence as well as disappointing detection performance for small targets. In this paper, we have improved encoder and decoder respectively for the above two issues. Firstly, we introduce multiscale encoder with feature interaction, in which there are only a few CNN operations. Additionally, we improved content object query and positional object query in the self-attention of decoder via introduce ground truth label embedding and dynamic anchor bbox, respectively. As result, it leads to impressive performance 46.9%AP and 28.8%AP_S on MS-COCO 2017 benchmark among the DETR-like detector using ResNet50 with DC5 or without DC5 of pre-trained on ImageNet as backbone trained in 50 epochs. We also conducted some experiments to confirm our analysis and verify the effectiveness of our method.

Keywords: Object detection, Transformer, CNN-free, Encoder, Decoder

1. Introduction

Object detection is a fundamental task in computer vision of wide applications. The object detection task is to predict the bounding box and class of each target in the image, and in the last decade, most detectors have been implemented based on deep convolutional networks. There are mainly two kinds of branch detection method, anchor-bbox method, such as RCNN family [1-2], SSD [3], YOLOv2-v5 [4-7], RetinaNet [8] as well as EfficientDet [9], and anchor-free method, including CenterNet [10], CornerNet [11], FCOS [12], and so on.

In contrast to anchor-based detectors, DETR [13] models object detection as a set prediction problem and uses 100 or 300 learnable object queries to probe and capture features from the output of Transformer encoders rather than from the neck in anchor-based method. And finally uses the binary graph matching algorithm to perform set-based box prediction and classification. Such a design effectively eliminates hand-designed anchors and non-maximum supperession (NMS) in the post-processing and makes object detection end-to-end optimizable. Since the input feature maps in encoder tend to be single scale, this is not friendly for detecting small targets or targets with large size differences. Besides, the positional object query in self-attention of decoder is randomly initialized and content object query usually initialized to a zero tensor, which cause plenty of time consumption for object query to capture independent areas of the image. To obtain a good performance, it usually takes 500 epochs of training on the COCO dataset, in contrast to 108 epochs used in the original Faster-RCNN-FPN training.

Much works has tried to identify the root cause and introduce multi-scale structure in the encoder or mitigate the slow convergence issue. Some of them address the problem through improving the model

[+] Corresponding author. Tel: +86 551 65108582.
E-mail address: dingzhuanlian@163.com

architecture. For instance, deformable DETR [14] replaces the global self-attention and cross-attention in encoder and decoder with a variability Attention structure. As for a query, each layer samples k points as a key, and uses a multi-scale encoder structure. Nevertheless, deformable DETR does not use the multi-scale fusion mechanism that combine FPN with MSA in encoder. Although SMCA [15] uses multi-scale encoder, it only replaces self-attention with FPN [16] in encoder directly, ignoring self-attention's ability to capture global features. As for Anchor DETR [17], positional object query and content object query are obtained based on randomly initialized 2D reference point and patterns through tensor's operations, so that each object query can capture image regions of 3 different pattern independently of each other. DAB-DETR [18] interprets the object query as a 4-D anchor box, integrates the scale information of anchor boxes into the object query on the basis of fusing the center point information, and updates it iteratively layer by layer.

Despite all the progress, few work pays attention to not only the effective encoder design about multiscale feature fusion and the bipartite graph matching part for more efficient training. In this study, we use swin-transformer as the backbone of model and design an encoder with a multiscale feature fusion mechanism rather than use FPN directly to replace the encoder in the DETR. Besides, we find that the slow convergence issue also results from the discrete bipartite graph matching component and the positional object query with no regularity. It lead to the whole train process of DETR is unstable especially in the early stages due to the nature of stochastic optimization most likely. As a consequence, for the same image, it is usually unstable for each object query to match the corresponding target, which makes optimization time-consuming and inconstant.

To address the problems above, in this paper, we propose a novel fully transformer detector with multiscale encoder and dynamic decoder. The main contributions are as follows:

1. We redesigned the encoder in DETR, three feature maps of different scales from swin are used as the input of encoder to carry out feature fusion before attention block. In this way, the advantages of attention and feature fusion can be fully utilized. Therefore, it can not only achieve cross-scale feature fusion, but also enforce cross-window information interaction under the same scale.

2. We analyze the root causes of sluggish DETR training process and gain a deeper understanding about DETR in training phase. We have made improvements to content query and positional query respectively. Content query continues to be obtained by the denosing training approach. Another adopts a parallel approach to integrate center point and scale information from 4-D anchor prior into content object query.

3. Experiments show that this method has a significant improvement in detection performance compared with original DETR, especially in the detection accuracy of small targets (+3.6AP and +6.3AP_S compared with ResNet50-DC5 as backbone). In addition, we also performed extensive ablation experiments to analyze the effectiveness of different components in our proposed model, for instance, mulitiscale encoder, encoder layers and scale of feature fusion.

2. Related Works

2.1. CNN-based Object Detection

With the emergence of deep convolutional networks, a large number of algorithms for object detection have emerged in the past few years. Although different subdivided detection tasks have appeared, such as domain adaptive detection, few shot detection and unsupervised or semi-supervised detection, anchor-based method is still in the dominant position. Anchor-based method is subdivided into two categories: anchor-bbox and anchor-point. Anchor-bbox detector is based on a series of prior boxes (proposals) manually set or obtained through clustering, finetune is performed according to the proposal, and the final detection result is obtained after post-processing. Among them, it can be divided into two-stage detector and one-stage detector according to the existence of RPN network. The two-stage detector family includes Fast RCNN [1], Faster-RCNN [2], Mask RCNN [19], Cascade R-CNN [20], Libra R-CNN [21], etc. As for the one-stage test family, the most representative ones are SSD [3], YOLOv2-v 5[4-7], RetinaNet [8], EfficientDet [9]. And another branch based on the anchor-free method is also developing rapidly, including: YOLOv1 [22], CornerNet [11], CenterNet [10], FCOS [12], etc.

2.2. Transformer and Its Variants

Since ViT [23] has been successfully applied to image tasks, the development of Transformer in image is becoming more and more rapid, and the Transformer system in visual tasks is becoming rich increasingly. The reason why Transformer attracts a large number of workers to study is that Transformer has a unified form. This unified form allows researchers to focus on more detailed design and innovation. It can be divided into the following two branches.

Backbone based on Transformer. Since the patch size segmented by each level of ViT is the same, that is, coarse-grained image patch is used as the input, it can be found that the network result of ViT is a columnar structure, rather than a trapezoidal structure like the classic CNN model such as ResNet [24]. Although ViT has achieved considerable performance in image classification tasks, the performance of semantic segmentation for high-resolution (e.g.800×1333) intensive image tasks is not good, mainly because the feature output of each level is single-scale and low-resolution. In addition, the computations and memory consumption brought by traditional self-attention is also huge. Many backbones based on transformer focusing on multi-scale and novel self-attention mechanism have emerged recently. For example, PVT [25] introduce different sizes of the patch embedding layer to ensure that the resolution of the feature map decreases successively, and the dimension of the feature map increases gradually. In addition, subsampling of key and value is done before self-attention. Swin [26] is also introduced as a hierarchical Transformer. It is also suggested that alternating the use of window MSA and shifted-window MSA not only reduces the computations of self-attention, but also brings cross-window interaction that limits attention to non-overlapping local windows, thereby pulling in richer context information and images features.

Transformer Detector. Applications of Transformer for CV tasks include: object detection, semantic segmentation, object tracking and so on. Here, we only analyze Transformer in the field of object detection. After the appearance of DETR [13] a novel object detection structure is gradually established. Due to the binary graph matching algorithm is used to calculate loss in the training process, the NMS operation is completely discarded in the post-processing, and it is a complete anchor-free and end-to-end detector. The drawback is that the query part of decoder in DETR is initialized randomly. As a result, it will take a lot of time for the object query to capture the independent and meaningful areas of the image. This is why DETR’s training time is too slow and difficult to convergence. SMCA [15] introduces a Gaussian prior to the cross-attention of decoder to adjust the learned attention weights map. Although the performance has been improved, it does not give a reasonable explanation for the slow training caused by DETR’s query. Conditional DETR [27] generates learnable positional queries by linear mapping of the position coordinates of the reference points, which participate in the training of cross-attention with the output of the encoder, decoupling the attention formula. Anchor DETR [17] obtains object query and decoder embedding based on the randomly generated reference point and pattern, so that each object query can independently capture different pattern image regions. As well as a novel attention mechanism in cross-attention of decoder is introduced to reduce computation and memory consumption. DN-DETR [28] consists of two parts: flip the real label to any other label at random with a certain probability and add box noise in two dimensions of center shifting and box scaling. Besides, in self-attention of decoder need to add an extra attention mask to prevent information leakage. While above works correlates object queries with location information, most of the work is based on low-resolution single-scale feature map for location search, ignoring a crucial problem: low-resolution single-scale detection is unfriendly to small targets and large size differences.

3. Methodology

3.1. Multiscale Encoder

DETR and a series of variants always selects the feature map that is the last bottleneck of the backbone as the input of the encoder and its corresponding positional encoding: $\mathrm{F} \in \mathbb{R}^{\frac{\mathrm{H}}{32}\times\frac{\mathrm{W}}{32}\times 2048}, \mathrm{Pos} \in \mathbb{R}^{\frac{\mathrm{H}}{32}\times\frac{\mathrm{W}}{32}\times 256}$. Why does our model use swin transformer as backbone instead of ResNet? The advantage of swin transformer lies in the hierarchy, locality and translation based on ViT By introducing the attention mechanism of alternate use W-MSA as well as SW-MSA, it achieves better performance in visual tasks compared with CNN-based networks. The specific advantages lie in the following two aspects: (1) **Hierarchy.** It can be seen from the original paper that the image is imported through patch partition and passes through multiple stages, in which patch mergeing and swin transformer block are processed respectively in each stage. In this process, the size of the image is constantly reduced. At the same time, the number of channels is constantly increasing, which is the hierarchy of the structure. This is why we chose swin transformer as backbone rather than ViT. (2) **Dependencies between Data.** The features learned from the attention mechanism are interrelated with each other with better universality and do not completely depend on the data itself. The focus is not only on local information, there is a diffusion mechanism from local to global to find expression. In addition, many parameters learned from CNN are static and fixed after learning, while Transformer can dynamically change each parameter. Meanwhile, transformer is more adaptable to big data such as COCO [29], and it is obvious that the performance of the model gets better and better as the amount of data increases.

Due to the large size difference of targets in COCO dataset, encoder and decoder only focus on single scale feature map for relationship modeling which is friendly for large target detection and the performance of small target detection will be degraded. Therefore, introduce multiscale encoder can not only solve the performance problem of large scale's gap between detection targets, but also make the detection of small targets more accurate. Unlike SMCA, it only replaces self-attention with FPN [16] in encoder directly, ignoring self-attention's ability to capture global features. In this case, key and value are completely based on the features extracted by CNNs, which loses the most important advantage of attention: message passing and the acquisition of global features, and based on this, using object query to match relevant regions and extract features for prediction may bring negative effects. Different from these methods, our approach is to use the idea of feature fusion for self-attention.

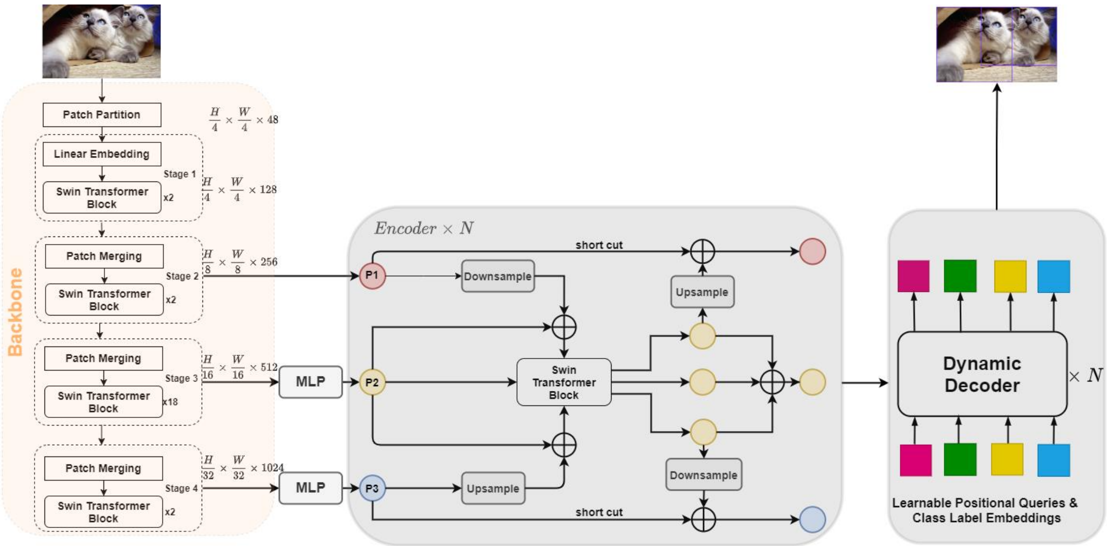

Fig. 1: Fully Transformer Detector architecture.

The overall network architecture is shown in Fig. 1. In encoder, feature maps P_1, P_2 and P_3 of the last three stages in swin transformer-Base are used as the input of encoder: $P_1 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 256}$, $P_2 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 512}$ and $P_3 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times 1024}$. And the downsampling factors relative to the original input image are 8, 16 and 32 respectively. Firstly, both feature map P_2 and P_3 go through an MLP block, which can be regarded as a 1x1 convolution operation, and the channel will be reduced to 256, which is convenient for subsequent feature fusion and self-attention operation. Then, self-attention is only used after each feature fusion. Of course, the attention mechanism here continues to use swin transformer block, namely W-MSA and SW-MSA. Feature map P_1 will through a downsample block, while feature map P_3 will through an upsample block, and carry out feature fusion with feature map P_2 respectively. Meantime, each fusion is followed by layer normalization and GELU activation functions. Now, there will be three different scale feature maps for self-attention. Finally, the three output feature maps in attention block are fused again to get one of the outputs of the current encoder block; The other two parts, contrary to the previous methods of upsampling and downsampling, go through the upsample and downsample block respectively, and conduct a feature fusion with the initial input P_1 and P_3 respectively. This process can be regarded as a short cut. Therefore, the above process is one multiscale encoder operation.

In the further analysis, for the feature map P_1 , P_2 and P_3 of three different scales, the semantic information and spatial location information contained in the feature map P_3(downsapling factor is 32) are the most critical for the CV task including image classification, object detection and semantic segmentation. No matter for W-MSA or SW-MSA attention mechanism, the above two types will involve the window division of feature map. For detection tasks, the size of input image is not fixed, and the size is not completely an integer multiple of 2, which leads to the feature map of the output of backbone cannot be divided into the same size windows. An additional padding operation is required to divide the window, and once the padding is done, the corresponding positional encoding also needs to be done. In this way, the original semantic information and spatial location information of P_3 may be destroyed. So, we don't enable self-attention operation to the last scale feature map. Besides, if P_1 is selected as the input standard of self-attention, the parameters and computations of the model will increase dramatically. Based on the above considerations, we finally choose the feature map of the intermediate scale as the standard of feature fusion. For the ablation experiment in this part, see Table 3.

3.2. Dynamic Decoder

In contrast to the self-attention of encoder and the cross-attention of decoder in DETR, the biggest difference between the above two branch is that the composition of query is different. Meanwhile, the positional encoding information of object query in DETR is randomly initialized, then reference point is obtained by object query through MLP and is not updated layer by layer, but a fixed mode prior center point. In addition, only the prior of anchor center point is generated, without the width and height information. Therefore, in the early training stage, the specific areas concerned by object query are either extremely large or extremely small, and it is difficult to capture the specific region in the same image. It takes a long time to train an object query to focus on a particular region. On the other hand, as for the content query, it is initialized to zero’s tensor in the DETR. Then, it was fused with the positional query and sent to the decoder's self-attention module. As a result, the decoder embedding will be mapped to the same space of feature map after the first cross-attention, which will also affect the convergence speed of the model.

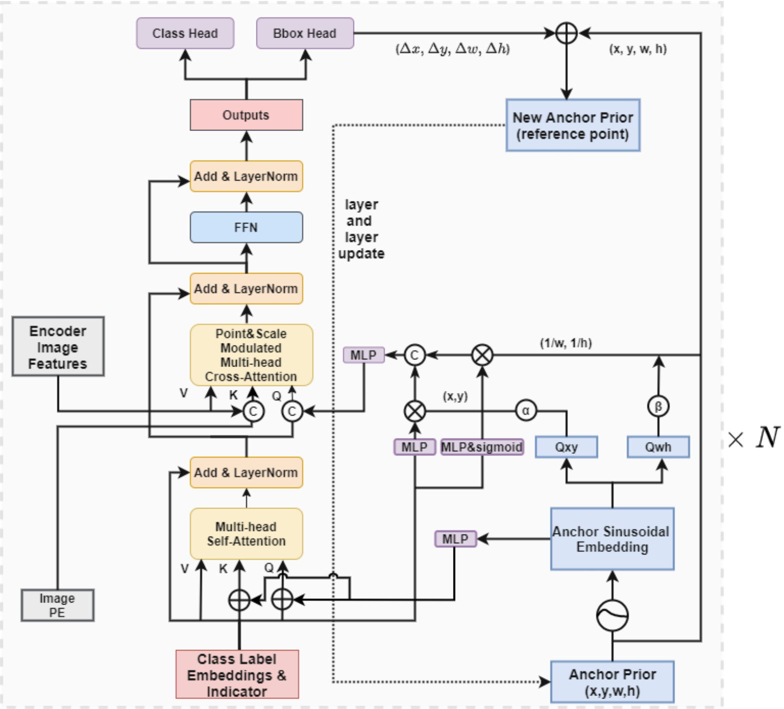

Fig. 2: Dynamic decoder structure.

For the convenience of explanation, we make the following declare: assume that $P_i \in \mathbb{R}^{N\times 4}$ represents the i-th anchor bbox, and the four dimensions respectively represent the parameter information of bboxes x_i, y_i, w_i, h_i . $Q_c \in \mathbb{R}^{N\times D}$ and $Q_p \in \mathbb{R}^{N\times D}$ represent content object query and positional object query, respectively. Here, D (e.g.256) represents the dimensions of the object query and decoder embedding, as for N represents the number of object queries. The overall structure of decoder is shown in Fig. 2.

Based on the thought proposed in Anchor DETR, object query should learn different patterns. Therefore, anchor prior through dimension transformation to obtain $A_i \in \mathbb{R}^{N \times p \times 4}$, here p represents the number of patterns to be learned.

$$A_i = \mathrm{ShapeModify}(\mathrm{P_i}) \tag{1}$$

$$\mathrm{tgt} = \mathrm{ShapeModify}(\mathrm{Pattern}) \tag{2}$$

The above equation 1 and 2 are expressed tensor operation of unsqueeze and repeat in PyTorch, which does not add extra any parameters or computations. Similarly, content object query of self-attention in decoder is tgt $\in \mathbb{R}^{\mathrm{N \times p \times D}}$ obtained by dimensional adjustment.

In the case of positional query generation in self-attention, the operation is similar to that in DAB-DETR. But the biggest difference in our approach is the positional query generation in cross-attention. In DAB-DETR, anchor boxes are operated by the sine embedding mapping to get object sine query, named Q_{base}. In order to facilitate the integration with decoder embedding as well as subsequent width height modulated. In the source code, Q_{base} directly intercepts the dimension and then performs subsequent operations. Equation 4 is as follows: $Q_{base} \in \mathbb{R}^{\mathrm{N \times 2D}} \rightarrow Q'_{base} = Q_{base}[\dots, :D] \in \mathbb{R}^{\mathrm{N \times D}}$

While the information contained in the first half of the original Q_{base} is generated based on the center point of anchor prior, the second half is generated based on the height and width of anchor prior. Only the embedding of the prior center point is obtained here, and then the integration with the decoder embedding on the center point dimension as well as the integration of the size information of anchor. Since the embedding of anchor point is used to integrate the previous decoder layer's output and scale information of anchor simultaneously, result in some embedding about anchor scale information may be lost. So, we adopt a parallel approach to integrate center point and scale information respectively. In order to better illustrate how we're fusing positional query and content query, here pattern will be set 1, batch size also be set 1. Firstly, $Q_{base} \in \mathbb{R}^{\mathrm{N \times 2D}}$ is generated in the same way as DAB-DETR. Here, $A \in \mathbb{R}^{N \times 4}$ indicates anchor prior.

$$Q_{\mathrm{base}} = \mathrm{SinEmbed}(\mathrm{sigmoid}(A)) \tag{3}$$

Next, we split the anchor embedding into two parts: center point and scale information. That is to say, Q_{base} is divided into $Q_{base}^{xy} = Q_{\mathrm{base}}[\dots, :D] \in \mathbb{R}^{N \times D}$ and $Q_{base}^{wh} = Q_{\mathrm{base}}[\dots, D:] \in \mathbb{R}^{N \times D}$ in the direction of the embedding dimension.

Meantime, we fused the positional query of the center point and the positional query of the scale information with the content query (tgt) respectively. In addition, two learnable hyper-parameters α and β will be used to adjust it respectively and dynamically maintain the embedded information corresponding to the learned center point and scale. Finally, the two fused parts $Q_{fused}^{xy} \in \mathbb{R}^{N \times D}$ and $Q_{fused}^{xy} \in \mathbb{R}^{N \times D}$ are concatenate and reduced by a linear mapping: $\mathrm{R^{2D}} \rightarrow \mathrm{R^{D}}$ to get positional query $\mathrm{Q_p}$ facilitate the fusion of content query $\mathrm{Q_c}$ in cross-attention module.

$$Q_{fused}^{xy} = \alpha \times Q_{base}^{xy} \times MLP(tgt) \tag{4}$$

$$Q_{fused}^{wh} = \beta \times \frac{Q_{base}^{wh}}{A[\dots,2:]} \times \mathrm{sigmoid}\big(MLP(tgt)\big) \tag{5}$$

$$Q_p = MLP\left(\mathrm{Concat}\big(\mathrm{Q_{fused}^{xy}}, \mathrm{Q_{fused}^{wh}}\big)\right) \tag{6}$$

In the summary, the improvement of decoder mainly includes two parts:

(1) **Positional Object Query in Self-Attention and Cross-Attention.** In the self-attention module, the content object query is obtained using a learnable pattern embedding; randomly initialize anchor prior to obtaining positional object query and positional key, respectively, by sine embedding function and MLP block with two linear mapping layers. In the cross-attention module, the content object query is the output from the encoder; The anchor prior embedding will be divided into the embedding of the center point and the embedding of the scale information in the feature dimension, and the feature fusion with the content object query will be carried out in the element-wise multiply mode. Then concatenate the two mapping results as a positional object query. In addition, the prediction output of the bbox head branch needs to obtain the reference point in element-wise additionally with the current anchor prior, which is used as the anchor prior of the next iteration so that the layer and layer update of anchor are realized.

(2) **Denosing Training Approach.** In essence, each layer of decoder will predict the relative offset $(\Delta x, \Delta y, \Delta w, \Delta h)$ and update the anchor box to get a more accurate anchor box$(\Delta x + x, \Delta y + y, \Delta w + w, \Delta h + h)$for prediction and transmit it to the next decoder layer. Therefore, the decoder can be thought of as learning two parameters: The position and relative offset of the anchor box. Besides, content queries in denoising training approach can be regarded as anchor position learning, while unstable matching of query and region will lead to predict unstable anchor, thus making relative offset learning difficult. For the content query, it is replaced by the label embedding, and the label is reconstructed by adding noise. For 4-D positional query, only adding a tiny positional disturbance near the ground truth as noise gives so that rebuilding the real box directly without the need for tedious Hungarian matching. Because denoising is a training mode that does not change the model structure, it is suitable for representing decoder query as 4-D coordinates. So, we adopt this trick as well.

4. Experiments

4.1. Experiment Settings

Dataset. We verify the proposed method in this paper on COCO2017[29] dataset. To be specific, model training is conducted on COCO2017 training set (118k images) and model verification is conducted on COCO2017's verification set (5k images).

Training Details. We continue to follow the training details of DETR. Backbone uses Swin-Base and Swin-Small, which are pre-trained on ImageNet [30] in original paper, and only finetune the last three stage's parameters. Weight decay set to 10^{-4}, using AdamW [31] optimizer, initial learning rate set to 10^{-3}. Learning rate scheduled to use StepLR: For training 50 epoch, the learning rate drops to 10 times of the initial value at the 40th epoch; For training 100 epoch, the learning rate drops to 10 times of the initial value at the 75th epoch and for training 150 epoch, the learning rate drops to 10 times of the initial value at the 100th epoch. Unlike DETR, 300 object queries were used instead of 100, and Generalized focal loss was used instead of the previous focal loss. The learning rate in encoder-decoder of DETR is set to 10^{-3} and the dropout factor ratio is 0.1. For multiscale encoder, feature map downsampling ratio is 8,16,32. All downsampling block adopt the feature fusion of convolution of $3 \times 3, s = 2$ and max pooling of 2×2. In the MSA module, we still use swin-transformer block. Related details about the loss matrix for binary graph matching: the coefficients of focal loss for classification, L1 distance loss and GIoU loss [32] for location are set as 2, 5, 2 respectively. After the best matching pairs of object query and GT were obtained, the whole network was trained by minimizing generalized focal loss [33], L1 loss and GIoU loss and the weights of the three loss parts were still 2, 5, 2. In addition, for data preprocessing, a more conventional data enhancement method is used: resize the image size ensure the minimum size of the short side of the image is 480px, the maximum size of 800px, and the maximum size of the long side of the image is 1333px; Randomly crop the original image at a ratio of 0.25; Flip horizontally at a random rate of 0.25. All of the model experiments were trained on four NVIDIA GeForce RTX 3090 with the batch size set to 1 on each GPU.

4.2. Main Results

We comprehensively compare the proposed fully transformer detector (**FTD**) with DETR and other variants of DETR, including single-scale and multi-scale models. The related DETR models are named as DETR-R50, DETR-R50-DC5, DETR-R101, DETR-R101-DC5 (R50 and R101 indicate that ResNet50 and ResNet101 are used as the backbone, respectively. DC5 indicates that the backbone uses dilated convolution in its last bottleneck) for the DETR variants. Our models are named FTD-Swin-B and FTD-Swin-S respectively. Table 1 shows the main results of COCO 2017 verification set. We compare the original DETR, Conditional DETR, Faster RCNN, Anchor-DETR, SMCA, Deformable DETR, DAB-DETR and DN-DETR. Compared with vanilla DETR using R101-DC5 as backbone achieved comparable performance in the 50 epochs training schedule: 46.9%AP vs 44.9%AP. Especially in the performance of detecting small targets, there is a big increase: 28.8%AP_S vs 23.7%AP_S. As for FFN module in attention block of encoder, the intermediate dimension is set to 1024, and the encoder layer is set to 4. The ablation experiment of encoder layers can be seen in Table 4.

Table 1: Comparison with DETR-like models and CNN-based model on COCO2017 validation dataset.

Model	Epoch	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params	GFLOPs	Multi-Scale
DETR-R50[13]	500	42.0	62.4	44.2	20.5	45.8	61.1	41M	86	
DETR-R50-DC5[13]	500	43.3	63.1	45.9	22.5	47.3	61.1	41M	187	
Conditional-DETR-R50[27]	50	40.9	61.8	43.3	20.8	44.6	59.2	44M	90	
Conditional-DETR-R50-DC5[27]	50	43.8	64.4	46.7	24.0	47.6	60.7	44M	195	
Anchor-DETR-R50[17]	50	42.1	63.1	44.9	22.3	46.2	60.0	39M	-	
Anchor-DETR-R50-DC5[17]	50	44.2	64.7	47.5	24.7	48.2	60.6	39M	-	
Deformable DETR-R50[14]	50	43.8	62.6	47.7	26.4	47.1	58.0	40M	173	√
Faster RCNN-FPN-R50[2]	108	42.0	62.1	45.5	26.6	45.5	53.4	42M	180	√
DAB-DETR-R50[18]	50	42.2	63.1	44.7	21.5	45.7	60.3	44M	84	
SMCA-R50[15]	50	43.7	63.6	47.2	24.2	47.0	60.4	46M	152	√
DN-DETR-R50[28]	50	44.1	64.4	46.7	22.9	48.0	63.4	44M	94	
FTD-Swin-S(ours)	50	**45.3**	**65.7**	**48.4**	**27.5**	**49.3**	**64.6**	84M	274	√
DETR-R101[13]	500	43.5	63.8	46.1	21.9	48.0	61.8	60M	152	
DETR-R101-DC5[13]	500	44.9	64.7	47.7	23.7	49.5	62.3	60M	253	
Conditional-DETR-R101[27]	50	42.8	63.7	46.0	21.7	46.6	60.9	63M	156	
Conditional-DETR-R101-DC5[27]	50	45.0	65.5	48.4	26.1	48.9	62.8	63M	262	
Anchor-DETR-R101[17]	50	43.5	64.3	46.6	23.2	47.7	61.4	58M	-	
Deformable DETR-R50[14]	150	45.3	64.3	49.1	27.1	48.4	60.0	40M	173	√
Faster RCNN-FPN-R101[2]	108	44.0	63.9	47.8	27.2	48.1	56.0	60M	246	√
DAB-DETR-R101[18]	50	43.5	63.9	46.6	23.6	47.3	61.5	63M	174	
SMCA-R101[15]	50	44.4	65.2	48.0	24.3	48.5	61.0	50M	218	√
DN-DETR-R101[28]	50	45.2	65.5	48.3	24.1	49.1	65.1	63M	174	
FTD-Swin-B(ours)	50	**46.9**	**66.8**	**49.8**	**28.8**	**50.6**	**65.8**	126M	445	√

4.3. Ablation Results

To validate different components of our proposed FTD, includes multiscale encoder and the fusion mechanism of positional object query. We perform a series of ablation studies in compared with the vanilla DETR. We choose original DETR with the ResNet101-DC5 backbone of the ImageNet-pretrained from TORCHVISION as baseline model for comparison. In order to ensure that the experiment is fair and reasonable, all experiments are trained for 50 epochs, the learning rate schedule and the optimizer is consistent, and the data preprocessing is the same.

Table 2: Encoder structure comparison. AP and AP_S are reported on COCO 2017 validation dataset. The multiscale encoder structure we proposed offer a nearly 2%AP increase compared to using the vanilla encoder.

Encoder structure	Epoch	AP	AP_S
Vanilla encoder[13]	50	45.1	24.3
FPN encoder[15]	50	45.8	26.2
Multiscale encoder	50	**46.9**	**28.8**

Multiscale Encoder. Combined with the previous analysis, we compare three different encoder structures: FPN encoder [15], vanilla encoder [13] and multiscale encoder proposed in our model, as show in Table 2 In order to ensure the fairness and effectiveness of the experiment, encoder layer 4 is adopted; Adaptive positional object query in decoder obtains the same way; Data enhancement, the hyper-parameters and learning plans of the training process are consistent. Swin-B is used as the backbone. Only the encoder structure was change, and all experiments were carried out on the COCO 2017 validation dataset.

Scale of Feature Fusion. In the feature fusion part of encoder. Which size feature map should be used as the fusion standard for the feature fusion before attention? Here, three different scale of feature map are mainly used for ablation experiments. The results are shown in Table 3. Because P_1 is selected as the input standard of self-attention, the parameters and computations of the model will increase dramatically.

Table 3: Ablation study on the scale of feature fusion, APand AP_S are reported on COCO 2017 validation dataset. The feature map with a subsampling ratio of 8 is used as the fusion standard, the rest are set up similarly.

Fusion scale	Epoch	AP	AP_S
P_1	50	43.8	24.1
P_2	50	**46.9**	**28.8**
P_3	50	46.1	27.7

Table 4: Ablation study on the encoder layers, AP and parameters are reported on COCO 2017 validation dataset.

Encoder layers	Epoch	Params(M)	AP
3	50	114	44.7
4	50	126	**46.9**
5	50	141	**47.1**
6	50	158	45.8

Encoder Layers. Since the encoder in our model includes both attention and feature fusion, encoder layer is an important factor affecting the model parameters, computations and the accuracy of detection. Similarly, keep the other model hyper-parameters, training parameters and data enhancement consistent. The results are shown in Table 4. When encoder layers is 3, it can be seen that the model parameters is small, but its detection accuracy is only 44.7%**AP** . It is not difficult to see that only three encoder layers cannot extract features well. However, when the encoder layer is 4, it is almost saturated. If the encoder layer continues to increase, there is only a small increase of **AP**, but the model parameters and FLOPs is huge. Surprisingly, precision even tends to drop if the same number of encoder layer as the DETR-like model is maintained.

Table 5: Ablation study on the way to obtain positional object query of cross-attention, AP and AP_{50} are reported on COCO 2017 validation dataset.

Positional object query	Epoch	AP	AP_{50}
learnable anchor point [13]	50	44.8	64.9
learnable anchor bbox [15]	50	46.3	66.4
separated learnable anchor bbox	50	**46.9**	**66.8**

Dynamic Decoder. We compare three ways of generating the positional object query of cross-attention. Since only changing the positional object query generation method brings about very small parameters and calculations, the comparison of the model parameters is not considered here. The results are show in Table 5. Learnable anchor point means that the original DETR method is used to directly treat the initialized object query as positional object query of cross-attention and the anchor point is not updated layer by layer. Besides, it only generates the anchor center point without introducing scale information. Learnable anchor bboxes represents the method proposed in DAB-DETR: 4-dimensional learnable boxes are obtained via sine function and MLP block, and then positional object query is gained by the fusion of center point and size information. Separated learnable anchor bboxes is the method we use to separate the center point and scale, and then conduct corresponding information fusion to obtain positional object query. The other model settings remain the same. Due to we used the denoising train approach in decoder so that the baseline we compared also used it.

5. Conclusion

We show a simple and efficient DETR variant with multiscale encoder and dynamic decoder. The multi-scale encoder performs the attention operation while the feature fusion is performed. This kind of clever involvement can not only make full use of the advantages of feature fusion, but also give play to the advantages of attention mechanism. Besides, we have also carried out decoder improvements to further explore the role of positional object query in cross-attention, and adopt a parallel approach to integrate anchor prior's center point and scale information into content object query. This shrinks the spatial range for the content object query to localize the distinct regions, thus capturing the RoIs more easily and reducing the training difficulty. The proposed detector can achieve better performance than the DETR and its variants.

Acknowledgement

This work was supported by NSFC (62076005, 61906002, U20A20398), NSF of Anhui Province (2008085MF191, 2008085QF306, 2008085UD07), and the University Synergy Innovation Program of Anhui Province (GXXT-2021-002).

6. References

[1] R. B. Girshick, Fast r-cnn, 2015 IEEE International Conference on Computer Vision (ICCV) (2015) 1440–1448.

[2] Shaoqing Ren and Kaiming He and Ross B. Girshick and Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2015 IEEE Transactions on Pattern Analysis and Machine Intelligence.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, 2015.

[4] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, 2016 CVPR.

[5] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, ArXiv abs/1804.02767.

[6] A.Bochkovskiy, C.-Y. Wang, Yolov4: Optimal speed and accuracy of object detection, ArXiv abs/2004.10934.

[7] U. L. Session, https://github.com/ultralytics/yolov5.

[8] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, P. Doll ́ar, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2017) 318–327.

[9] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, 2020 CVPR.

[10] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, 2019 ICCV.

[11] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, 2019IJCV 128 (2018) 642–656.

[12] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, 2019 ICCV.

[13] N. Carion, F. Massa, G. Synnaeve, End-to-end object detection with transformers, 2020 ECCV.

[14] X. Zhu, W. Su, Deformable detr: Deformable transformers for end-to-end object detection, arXiv:2010.04159.

[15] P. Gao, M. Zheng, X. Wang, Fast convergence of detr with spatially modulated co-attention, 2021 ICCV.

[16] T.-Y. Lin, P. Doll ́ar, R. B. Girshick, K. He, Feature pyramid networks for object detection, 2016 CVPR.

[17] Y. Wang, X. Zhang, T. Yang, J. Sun, Anchor detr: Query design for transformer based object detection, 2021.

[18] S. Liu, F. Li, H. Zhang, X. Qi, H. Su, Dab-detr: Dynamic anchor boxes are better queries for detr, 2022 ICLR.

[19] K. He, G. Gkioxari, P. Doll ́ar, R. B. Girshick, Mask r-cnn, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2017) 386–397.

[20] Z. Cai, N. Vasconcelos, Cascade r-cnn: Delving into high quality object detection, 2017 CVPR.

[21] J. Pang, K. Chen, J. Shi, H. Feng, Libra r-cnn: Towards balanced learning for object detection, 2019 CVPR.

[22] J. Redmon, S. K. Divvala, R. B. Girshick, You only look once: Unified, real-time object detection, 2016 CVPR.

[23] A.Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, An image is worth 16x16 words: Transformers for image recognition at scale, 2021 ICLR.

[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015 CVPR.

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

[26] Z. Liu, Y. Lin, Y. Cao, Swin transformer: Hierarchical vision transformer using shifted windows, 2021 ICCV.

[27] D. Meng, X. Chen, Z. Fan, G. Zeng, Conditional detr for fast training convergence, 2021 ICCV.

[28] F. Li, H. Zhang, S. guang Liu, Dn-detr: Accelerate detr training by introducing query denoising, 2022 CVPR.

[29] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll ́ar, C. L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, 2014.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009) 248–255.

[31] Loshchilov, F. Hutter, Fixing weight decay regularization in adam, ArXiv abs/1711.05101.

[32] S. H. Rezatofifighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, 2019 CVPR.

[33] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, J. Yang, Generalized focal loss: Learning qualifified and distributed bounding boxes for dense object detection, ArXiv abs/2006.04388.