Neural Computing and Applications (2024) 36:6793–6808
https://doi.org/10.1007/s00521-024-09430-6

ORIGINAL ARTICLE

FCDS-DETR: detection transformer based on feature correction and double sampling

Min Wang[1] · Zhiqiang Jiao[1] · Zhanhua Huang[2] · Shihang Yu[3]

Received: 6 August 2023 / Accepted: 15 January 2024 / Published online: 9 February 2024

Abstract
The recently proposed semantic-aligned matching detection transformer (SAM–DETR model) accelerates the convergence of the detection transformer (DETR) by mapping object queries into an identical embedding space as the encoder's output feature map. However, SAM–DETR model has the problem of low detection accuracy compared to other DETR variants. We observe that the lower detection accuracy of SAM–DETR model is caused by the insufficient number of sample points and the inaccurate localization of the sample points during re-sampling, which blurs the generated attention map. This paper proposes an object detector based on a feature correction and double sampling DETR (FCDS-DETR) to solve this problem. FCDS-DETR takes SAM–DETR model as a baseline and builds on it by adding a feature correction module and a double sampling mechanism to achieve further improvement in detection accuracy with a limited number of additional parameters without sacrificing convergence speed. Firstly, FCDS-DETR improves the sampling point localization accuracy by adding a feature correction module to model the inter-channel dependence of the feature maps to be sampled. Secondly, the number of sampled points is increased by the double sampling mechanism, and attention fusion is used to fuse the attention weight maps corresponding to the two sets of sampled points to improve the recognizability of the attention weight maps. The experimental results show that the average precision is improved by +0.7 on the COCO dataset compared with the SAM–DETR model, and the number of parameters is increased by only 10.34%, which improves the detection performance of the model very well.

Keywords Object detection · DETR · Feature correction · Double sampling

1 Introduction

Object detection is one of the fundamental tasks of computer vision and aims to predict a set of bounding boxes and category labels for each object instance of interest [1]. According to the process of object bounding boxes from nothing to something, we can divide into single-stage and two-stage object detection. The two-stage object detection algorithm in the first stage mainly uses anchors, region proposals, and NMS [2] to find out where the object appears and get suggestion bounding boxes. The second stage uses classifiers to classify the suggestion bounding boxes and finally realize the object detection process. The R-CNN [3–6] family is the classical two-stage object detection algorithm, among which fast R-CNN [7] and faster R-CNN [8] are regularly used in the area of object detection by virtue of their excellent capabilities. Although these detection algorithms have high detection accuracy, the detection speed is usually measured in frames per second, and even the fastest high-accuracy detector, Faster-R-CNN, can only run at 7 frames per second (FPS), making it unsuitable for detection scenarios with high time response requirements [9].

In order to solve the above problem, single-stage object detection has emerged. It requires only one forward pass of a single neural network model to predict objects' class and location information directly from the original image.

✉ Zhanhua Huang
zhanhua@tju.edu.cn

[1] School of Control Science and Engineering, Tiangong University, Tianjin 300387, China

[2] School of Precision Instrument and Opto-Electronics Engineering, Tianjin University, Tianjin 300072, China

[3] School of Mechanical Engineering, Tiangong University, Tianjin 300387, China

Classical single-stage object detection algorithms include YOLO [10–15], SSD [9], ResNet [16], etc. These classical object detection algorithms must count on manual components to accomplish the object detection process and cannot achieve true end-to-end object detection.

With the continuous exploration of researchers in object detection, Carion et al. [17] proposed an object detector that regarded object detection as a direct set prediction problem, named detection transformer (DETR), which brought a new direction for the development of object detection. DETR is based on encoder–decoder architecture [18] and combined with the bipartite graph matching algorithm, which eliminates the dependence on many manual components and realizes end-to-end object detection. It effectively simplifies the process of object detection.

Although DETR has a simple structure and achieves end-to-end object detection without relying on manual components, its training and inference still take a long time. The effect is not very good when detecting small- and medium-sized objects. Therefore, much subsequent research on DETR has been devoted to speeding up its convergence and improving detection accuracy.

Zhu et al. [19] attribute the slow convergence of DETR to the fact that the attention module applies the same attention weights to all pixels in the feature map in the initial stage. It causes the model to take a lot of time to learn the object distribution in the dataset to be detected. Based on the above analysis, deformable DETR is proposed. The problem of slow convergence of DETR is alleviated. Meng et al. [20] concluded that the slow convergence of DETR is because the query's content embedding must be matched with both the content embedding in the key and the spatial embedding in the key when computing cross-attention by visualizing the spatial attention weight map of cross-attention in DETR. Therefore, DETR needs a large number of epochs to improve the quality of content embedding to locate the object precisely. Based on the above analysis, the authors propose conditional DETR. By decoupling the cross-attention module in the decoder, the convergence speed of DETR is improved. Gao et al. [21] propose a plug-and-play spatially modulated cross-attention module named SMCA and apply it to DETR. SMCA-DETR introduces a 2D spatial Gaussian-like distribution in the cross-attention mechanism. In this way, the search range of each object query vector in cross-attention is adjusted to a certain distance close to the target center, thus accelerating the convergence speed of DETR. In addition, the authors integrate multi-head attention and scale-selective attention into SMCA to further improve the detection accuracy of the model. Zhang et al. [22] proposed a strong end-to-end object detection model DINO, which improves the slow convergence speed of DETR-like models and the unclear meaning of query vectors by using a contrastive denoising training method, a hybrid query selection mechanism, and twice forward propagation. In 2022, Zhang et al. [23] experimentally found that in the DETR decoder, the object queries were mapped multiple times in the self-attention module and FFN, resulting in a lack of semantic alignment between the object queries and image features, which affected the convergence of DETR. Based on this analysis, Zhang et al. proposed a semantic-aligned matching detection transformer (SAM–DETR model), a model to accelerate DETR convergence by semantic alignment matching. SAM–DETR model greatly accelerates the convergence of DETR without sacrificing accuracy.

Inspired by the effectiveness of the above multi-head attention [18] and semantic alignment matching re-sampling mechanism [23] in accelerating DETR convergence and improving model detection performance, we propose a DETR object detector based on feature correction and double sampling (FCDS-DETR). It improves the detection accuracy by improving the perception ability of the baseline model to the target object. Specifically, we add a feature correction module to SAM–DETR model, which indirectly affects the position of sampling points in the sampling area by explicitly modeling the inter-dependence between feature channels. The ability of the model to locate the edge and end of the detected object was enhanced. At the same time, the double sampling mechanism and feature map fusion method of FCDS-DETR can improve the recognizability of the attention weight maps and reduce the difficulty of subsequent matching tasks by fusing the attention maps generated by the two sets of sampling points. The specific contributions of this paper are as follows:

1. We propose a high-accuracy end-to-end object detector that utilizes the feature correction module and double sampling mechanism to enhance the SAM–DETR model's ability to detect and localize targets, thereby improving detection accuracy.
2. The reason behind the fuzziness of the attention map generated by the SAM–DETR model is thoroughly analyzed, and a novel attention fusion method is proposed to enhance the recognizability of the attention weight map and achieve more reliable object detection.
3. We evaluated our proposed model on the improved FDDB [24] and COCO [25] datasets, conducting a comprehensive assessment of its performance metrics, including precision, recall, and model parameters, through statistical measurements. The method is compared with existing DETR-like models such as deformable DETR and SAM–DETR model.

2 Related Work

2.1 Transformer

In the field of natural language processing (NLP), recurrent neural networks (RNNs) have long been one of the most popular neural network architectures. Still, they often suffer from long-term dependency problems when processing long sequences, leading to unsatisfactory results. In order to solve this problem, Vaswani et al. [18] proposed a new deep learning model transformer. Compared with RNNs, transformer is an attention mechanism-based neural network model that can model long sequence data without using the recurrent structure and has better parallel computing capability and computational efficiency. The success of transformer in the field of NLP has laid the foundation for subsequent applications in image classification [26–29], image generation [30–33], action recognition [34–36], and fault diagnosis [37, 38]. The transformer structure exchanges information with all inputs by employing key, query, and value. Through continuous iterative learning, it establishes the connection between each input and itself and the connection between each input and other inputs. However, the transformer structure is computationally quadratic in complexity, making it computationally intensive in the learning process and requiring a long time for model training. To address this problem, a series of studies have been conducted by subsequent scholars. Sparse transformer [39] uses sparse attention instead of the dense attention of the traditional transformer, reducing the complexity of the transformer from $O(n^2)$ to $O(nlog(n))$. Linformer [40] proposes to remove the softmax function in the transformer and perform matrix multiplication between query and value first to achieve the complexity from $O(n^2)$ down to $O(n)$. In this paper, our FCDS-DETR constructs the model by referring to the idea of the original transformer. In future work, we will explore efficient transformers in FCDS-DETR.

2.2 Siamese-based architecture for matching

Siamese-based architecture for matching, a deep learning model for similarity comparison and matching, has a main structure consisting of Siamese networks. By stitching two identical neural networks together for training, they are accelerated to learn to project the two input vectors into the same feature space. The model projects two input vectors into two new vectors through a neural network when performing similarity matching. The similarity between these two vectors is then determined by calculating the Euclidean distance in the embedding space and other methods. This method is widely used in text matching [41–43], image matching [44–46], and face verification [47–49]. Mueller [42] et al. proposed the Siamese Recurrent Architecture in 2016 and used it for text matching tasks. The advantage of this model is its ability to learn the representation of text adaptively and capture the semantic similarity between texts. A good generalization capability was obtained by training on a limited dataset. Chen [43] et al. proposed an enhanced Siamese-based architecture model for natural language inference tasks in 2017. The model is able to handle diverse text types and lengths efficiently. Florian [50] et al. proposed a method of training similarity from data and used it for face verification. Koch [51] et al. used Siamese networks for small-sample learning tasks, introducing distance metrics to solve the problem of small-sample classification. Our FCDS-DETR achieves semantic alignment matching by projecting object queries to the same embedded space as the encoder output feature map.

2.3 Classical feature fusion method in object detection

Feature fusion methods have been widely adopted in the field of object detection due to their superior performance, with the main goal being to enhance and optimize the model's detection accuracy for targets of varying scales and perspectives. In 2017, Lin et al. [52] proposed the feature pyramid network (FPN), an effective feature fusion architecture that uses a top-down pathway and lateral connections to integrate multi-scale and multi-level feature information, thereby improving object detection performance. Subsequently, PANet, proposed by Liu et al. [53], adds a bottom-up pathway to the FPN, allowing feature fusion to occur at each level and better integrating information from lower and higher layers. Huang et al. [54] designed DenseNet, which establishes dense connections between all layers, offering an effective method for integrating multi-scale and multi-level information, hence improving the model's generalization capability. In this paper, we draw insights from these feature fusion methods and apply them to the fusion process of cross-attention weight maps, on the basis of which we propose an attention fusion method. This innovative approach improves the detection performance of the model FCDS-DETR from a new perspective.

3 Proposed methods

3.1 Overview

The FCDS-DETR proposed in this paper aims to integrate the feature correction module and the double sampling mechanism into the Semantics Aligner module to improve

the problem of insufficient sampling points and inaccurate sampling point positioning in the re-sampling process of SAM–DETR model. The model's accuracy is enhanced while keeping the number of parameters limited and without sacrificing the convergence rate of the model. In the following sections, we first review the basic architecture of SAM–DETR model and then introduce the architecture of our proposed FCDS-DETR.

3.2 Review SAM–DETR model

SAM–DETR model uses ResNet-50 [16] as a feature extraction network to extract feature maps $F \in R^{H*W*C}$ from the input image $I \in R^{H0*W0*3}$, where $H0$, $W0$, and H, W represent the height and width of the input image and output feature map, respectively. C represents the dimension of the output feature map. In the encoder part, the feature map F is first combined with the sinusoidal position encoding PE to obtain the feature map F_{pe} containing spatial position information. The F_{pe} will be applied to generate vectors K and Q, representing Key and Query in the transformer, respectively. The sinusoidal position encoding PE formula is shown in Eq. (1) and Eq. (2),

$$PE(pos, 2i) = Sin(\frac{pos}{10000^{\frac{2i}{d}}}) \tag{1}$$

$$PE(pos, 2i+1) = Cos(\frac{pos}{10000^{\frac{2i}{d}}}) \tag{2}$$

where pos represents the position coordinate of each pixel of the feature map, $2i$ and $2i+1$ represent each pixel's position in the corresponding position embeddings, and d is the dimension of the position embeddings. V represents the value obtained from F without position information. K, Q, and V will be input to self-attention. In the calculation of self-attention, the matrix dot product between Q and K is used to obtain the output containing context information. And then, after normalization and linearization, we get the output of self-attention. Realize the information exchange between features in all spatial locations. To increase feature diversity, K, Q, and V are divided into groups along the channel dimension for *MHSAttention* (multi-head self-attention). This *MHSAttention* formula is shown in Eq. (3),

$$MHSAttention(Q, K, V) = Concat(Softmax(\frac{Q_i K_i}{\sqrt{d_k}})V_i)W^o \tag{3}$$

where Q_i, K_i, and V_i represent the ith feature groups of Q, K, and V. The d_k is the row dimension of Q_i and K_i. W^o represents the output transformation matrix. The output result of MHSAttention is transformed and input to the transformer's decoder.

In the decoder part, SAM–DETR model adds a semantics aligner before each multi-head cross-attention, as shown in Fig. 1. This Semantics Aligner samples the encoder output feature maps to generate new query, the generation of which is shown in Fig. 2. The gray ellipse represents the necessary input data for cross-attention, while the orange rectangle depicts the process of creating a new query vector. Subsequently, the cross-attention module takes new query as input. The semantic aligner ensures that key and query are semantically aligned in cross-attention since both key and query are derived from the SAM–DETR model encoder output feature maps. By adding a semantic aligner to DETR, the convergence of the model is accelerated, and the detection accuracy is improved.

It is because of SAM–DETR model's unique understanding of the problems of the classical DETR model that it opens up a new direction to accelerate DETR convergence. Using the re-sampling method to obtain object query avoids the problem of semantic destruction caused by multiple projections of object query generated by multiple decoder superposition. At the same time, the detection accuracy of the model is improved to a certain extent. However, the number of sampling points used by SAM–DETR model in the re-sampling process of semantics aligner is small, while the sampling points are not accurate

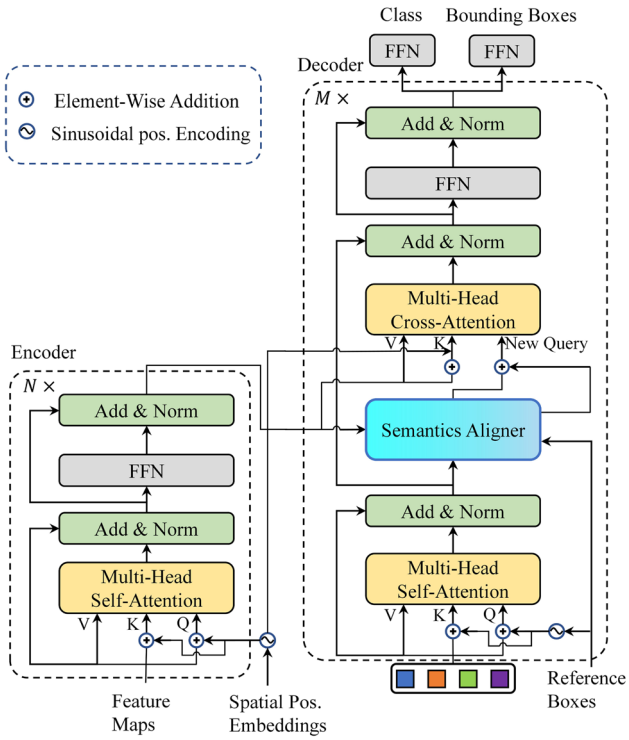

Fig. 1 Structure of SAM–DETR model. N denotes the number of encoders in the transformer, and M denotes the number of decoders in the transformer

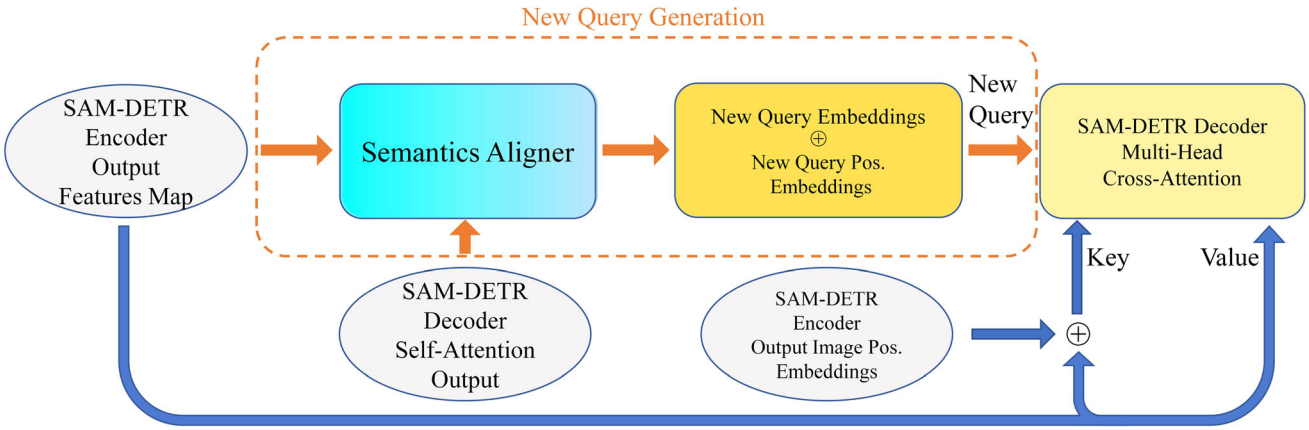

Fig. 2 Semantic aligner module generates new query vectors

enough in locating the crucial positions of target objects, resulting in low detection accuracy of the model.

3.3 Feature correction module

Feature correction is typically employed in feature extraction networks to enhance the robustness and generalization of the model by correcting and refining the features in the middle layer of the model. Since the transformer has the quality of multilayer decoder stacking, we introduce the feature correction module of parameter sharing into the basic decoder module. In a single iteration of the model, the multilayer stacked decoder can complete the recalibration of the input features by using the feature correction module and constantly updating the coordinates of the sampling points. CAM [55] is a classical squeeze-and-excitation type of feature correction mechanism, which can improve the quality of the feature representation of the model by adding only a few parameters. Therefore, we use it as a feature correction module to correct the features of the ROI extraction region. Our core idea is to indirectly affect the sampling points in the sampling area by using the differences between the channels of the feature map to be sampled. This indirect effect is mainly due to the fact that the feature map to be sampled is recalibrated in the channel dimension after adding the feature correction module. When the recalibrated feature map to be sampled is used to predict the offset of the re-sampled coordinate points, more accurate prediction results are obtained. The model's ability to perceive the area to be detected and locate the key points of the object is improved, which in turn leads to improved detection accuracy. As shown in Fig. 3, E is the feature map obtained by the transformer encoder image feature after convolutional transformation. E_{roi} is the potential region containing the object obtained by E after ROI align [56]. F_{sq} completes the squeezing operation of the global spatial dimension of the input features, which we implement using global max pooling and global average pooling. This operation describes the feature map channel dimension by aggregating the input feature map spatial dimensions. The F_{sq} formula is shown in Eq. (4),

$$F_{\text{sq}}(E_{\text{roi}}) = \begin{cases} \dfrac{1}{H_{E_{\text{roi}}} \times W_{E_{\text{roi}}}} \displaystyle\sum_{x=1}^{W_{E_{\text{roi}}}} \sum_{y=1}^{H_{E_{\text{roi}}}} E_{\text{roi}}(x,y) \\ Max(E_{\text{roi}}(x,y)),\ x \in W_{E_{\text{roi}}}, y \in H_{E_{\text{roi}}} \end{cases} \tag{4}$$

where $H_{E_{\text{roi}}}$ represents the height of the ROI align output feature map and $W_{E_{\text{roi}}}$ represents the width of the ROI align output feature map.

We input the global average pooling and the global max pooling results into the MLP to obtain two sets of $1 \times 1 \times C$ feature vectors. After that, we add the two sets of feature vectors and use the sigmoid function to implement the excitation operation F_{ex}. The weight set W_{roi} for each feature channel corresponding to the feature map E_{roi} is ultimately generated through the continuous iterative learning of the MLP. Figure 3 shows that F_{adject} multiplies the weight set W_{roi} with the original E_{roi} to obtain the recalibrated feature map E_{adj}. The F_{ex} and F_{adject} formulas are shown in Eqs. (5) and (6),

$$F_{\text{ex}}(X) = Sigmoid(MLP(X)) \tag{5}$$

$$F_{\text{adject}}(E_{\text{roi}}) = E_{\text{roi}} \times W_{\text{roi}} \tag{6}$$

where X represents the output of E_{roi} after the global max pooling and the average pooling.

3.4 Double sampling mechanism

The purpose of re-sampling is to find the most representative key points in the feature map containing potential objects. In the semantics aligner module of SAM–DETR model, the regions containing potential objects are first extracted using ROI align, and a single re-sampling is performed within the region. Then, the sampled results are

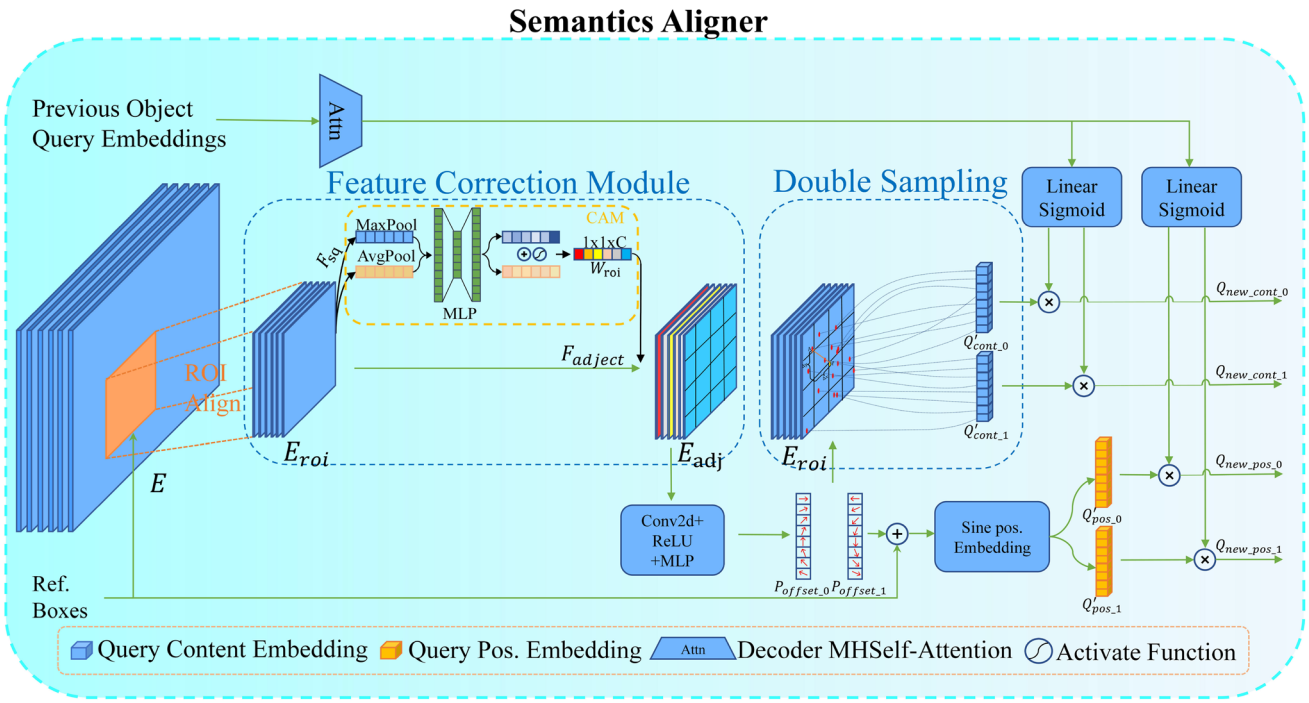

Fig. 3 Overall structure of semantics aligner used in FCDS-DETR, including the feature correction module and the double sampling mechanism. The feature correction module improves the sampling point localization accuracy by explicitly modeling the dependencies between the feature channels to be sampled. The double sampling mechanism samples on the feature region E_{roi} to generate the input of multi-head cross-attention

sent to multi-head cross-attention as object queries for the dot product calculation of attention. Finally, an attention weight map that reveals the matching degree between object queries and the target region is generated. However, while this single re-sampling scheme can speed up the convergence of DETR, it suffers from under-sampling when using fewer sampling points to locate the key points of the target object in the reference box.

Based on the above analysis, this paper proposes a double sampling mechanism to enhance the object perception capability of the model by increasing the number of sampling points, thus improving the detection accuracy. First, a feature map containing inter-channel dependencies is used to predict the offset of the two sets of sampling points. Then, the bilinear interpolation algorithm is used to double sample on the original feature map extracted by ROI Align according to the offset of the generated sampling points. Finally, two sets of query content embeddings are generated. Let us denote the two sets of offsets as P_{offset_i}, where $i = \{0, 1\}$ represents the group to which they belong. As shown in Fig. 3, We obtain the offset of the sampling points by passing the recalibrated feature map E_{adj} to Conv, ReLU activation, and MLP. The formula is shown in Eq. (7),

$$P_{\text{offset}\_i}(E_{\text{adj}}) = MLP(ReLU(Conv(E_{\text{adj}}))) \tag{7}$$

where Conv denotes the convolution operation, and ReLU is the activation function.

Following that, we can easily obtain two sets of corresponding key sampling points in the feature map to be sampled using an interpolation algorithm based on P_{offset_i}. The purpose of grouping is to be compatible with the number of multiple heads in cross-attention and to facilitate attention fusion in subsequent attention fusion in Sect. 3.6. We use F_{Ds} to represent the interpolation re-sampling. The Q'_{cont_i} represents the re-sampling output of the corresponding group. The formula is shown in Eq. (8).

$$Q'_{\text{cont}\_i} = F_{\text{Ds}}(E_{\text{roi}}, P_{\text{offset}\_i}) \tag{8}$$

P_{offset_i} is also used to update the coordinate boxes of the ROI regions and generate a new position embedding Q'_{pos_i}.

This paper does not discard the use of previous query embedding in the semantics aligner module. Instead, the number of weights generated by the linear projection is increased to match the output of the double sampling. The formula is shown in Eqs. (9) and (10),

$$W_{\text{pre}\_i} = \text{Sigmoid}(\text{Linear}(Q_{\text{pre}})) \tag{9}$$

$$Q_{\text{new\_cont}\_i} = W_{\text{pre}\_i} \times Q'_{\text{cont}\_i} \tag{10}$$

where Q_{pre} represents the previous query embedding, W_{pre_i} represents the weight value of the previous query embedding after linear projection and sigmoid function activation and $Q_{\text{new}\_\text{cont}\_i}$ represents the weighted query content embeddings.

3.5 Sampling point coordinate information acquisition and position embedding

Before performing sinusoidal position embedding on the sample points, the coordinate information of the sample points should be obtained according to the position offset predicted by the MLP network in Eq. (7). Compared with the offset prediction network of the original SAM–DETR model, FCDS-DETR chooses to double the output channels of the MLP in the last layer of the offset prediction network to achieve the prediction of two sets of offsets. As shown in Fig. 4, the yellow reference point $P(x\_\text{center}, y\_\text{center})$ is the center point coordinate of the ROI aligner output feature map E_{roi}, which is used as the reference point in the prediction of the model for the sample point. The red point $A(x_{\text{sp}}, y_{\text{sp}})$ is the sample point. Δx, Δy denote the offset of the sample point A relative to the reference point P in the x, y direction. $W_{E_{\text{roi}}}$ and $H_{E_{\text{roi}}}$ denote the width and height of the feature map E_{roi}. The formula for calculating the sampling point A is shown in Eq. (11),

$$\begin{cases} x_{\text{sp}} = x\_\text{center} + \Delta x \\ y_{\text{sp}} = y\_\text{center} + \Delta y \end{cases} \tag{11}$$

The formulas for Δx and Δy are shown in Eq. (12),

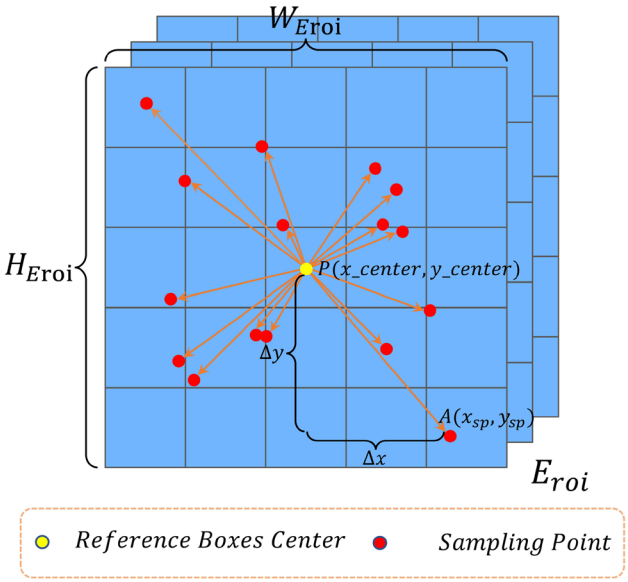

Fig. 4 Sampling point acquisition method

$$\begin{cases} \Delta x = \dfrac{W_{E_{\text{roi}}}}{2} \times P_{\text{offset}\_x} \\ \Delta y = \dfrac{H_{E_{\text{roi}}}}{2} \times P_{\text{offset}\_y} \end{cases} \tag{12}$$

where P_{offset_x} and P_{offset_y} represent the outputs of the MLP in the offset prediction network.

In this paper, we inherit the way of position embedding in DETR and perform sinusoidal position encoding on the position coordinates of sampling points to generate two sets of corresponding query position embeddings Q'_{pos_i}. We also increase the number of weights from the linear projection of previous query embeddings to generate new query position embeddings $Q_{\text{new}\_\text{pos}\_i}$.

3.6 Attention fusion

The cross-attention mechanism plays a crucial role in SAM–DETR model, which achieves target matching and feature extraction by using the sampled points from the encoder output feature map as object queries. However, the cross-attention weight map of SAM–DETR model is based on a single re-sampling, which is affected by the accuracy of sampling points, making the attention weight map blurred and unable to locate the target object precisely. Based on the above analysis, the attention fusion method is proposed in this paper. The two sets of sampled points obtained by the double sampling mechanism are fed into the cross-attention module in parallel, and the resulting cross-attention weight map is fused to improve the sensitivity and detection accuracy of the model on the target object.

We multiply the two sets $Q_{\text{new}\_\text{cont}\_i}$ and the corresponding $Q_{\text{new}\_\text{pos}\_i}$ with their respective weight matrices $W_{q\_\text{cont}\_i}$ and $W_{q\_\text{pos}\_i}$, respectively. After that, the two query vectors Q_i can be obtained by summing. The formula is shown in Eq. (13),

$$Q_i = (W_{q\_\text{cont}\_i} \times Q_{\text{new}\_\text{cont}\_i}) + (W_{q\_\text{pos}\_i} \times Q_{\text{new}\_\text{pos}\_i}) \tag{13}$$

where $W_{q\_\text{cont}\_i}$ and $W_{q\_\text{pos}\_i}$ represent the weight matrices obtained after linearization and sigmoid activation of the self-attentive outputs in the decoder. Q_i represents the new query vectors generated after the semantic alignment module. The grouping of object queries does not impact the key and value in MHCAttention (multi-head cross-attention). We represent the attention weight map obtained by multi-head cross-attention as F_{w_i}, with Eq. (14). After that, the weight map F_{w_0} with F_{w_1} overlay is used to achieve attention fusion.

$$F_{w_i} = MHCAttention(Q_i, K, V) = Soft\max(\frac{Q_i K^T}{\sqrt{d_k}})V \tag{14}$$

The fusion attention process is shown in Fig. 5, where (a) is F_{w_0}, (b) is F_{w_1}, and (c) is the cross-attention weight map generated after F_{w_0} and F_{w_1} are fused. Looking at subfigure (c) in Fig. 5, it is clear that the weight map, after attentional fusion, has richer boundary and content information compared to subfigures (a) and (b). This is particularly evident for the medium and small targets enclosed by the red box in the figure. Attention fusion helps to distinguish these targets from others, thereby reducing the training pressure on the subsequent FFN.

4 Experiment

4.1 Image dataset preparation

In this paper, experiments are conducted on the improved FDDB dataset and the COCO 2017 dataset. The improved FDDB dataset converts the elliptical face annotation of the original FDDB dataset to an external rectangular annotation and converts the interpretation file to COCO format to fit the model's requirements for the dataset. The improved FDDB dataset contains 2100 training images and 745 validation images with a total of 5171 face targets. The COCO 2017 dataset contains 118k training images and 5k validation images. Each image has an average of seven instances, and a single image in the training set contains up to 63 instances, each annotated with bounding boxes.

4.2 Experimental setup

For the improved FDDB dataset, we explore the model's performance at 100 epochs. The initial learning rate of the model is set to 1×10^{-5}, and the learning rate decreases to 1/10 of the initial value at the 80th epoch. The batch size is set to 8. For the COCO dataset, we first experiment with 12 epochs, which is widely used in the ConvNet detector [8], and second, we also experiment with 50 epochs based on the transformer detector [19, 20]. The initial learning rate for these two sets of experiments is set to 1×10^{-5}, and the AdamW [57] optimizer is used. The batch size is set to 6. We use a configuration of 4×Nvidia GeForce RTX 3090 GPUs to train our model, considering the requirement of GPU computing power for the transformer-based model in the learning process. When FCDS-DETR employs multi-scale in the encoder, the batch size is reduced to 2 due to the significant CUDA memory required. For the comparison experiment under 12 epochs, the model kept the initial learning rate unchanged. In the comparison experiment under 50 epochs, the learning rate decreases to 1/10 of the initial value at the 40th epoch. The input image size is set between 480*480 and 1333*1333 pixels, and the data are enhanced using random cropping and horizontal flipping.

4.3 Evaluation indicators

Both datasets we used for our experiments are in the annotated format of the COCO dataset. Therefore, we use the COCO evaluation indicators to evaluate the model's detection performance objectively. Among them, we pay more attention to the detection of average precision and

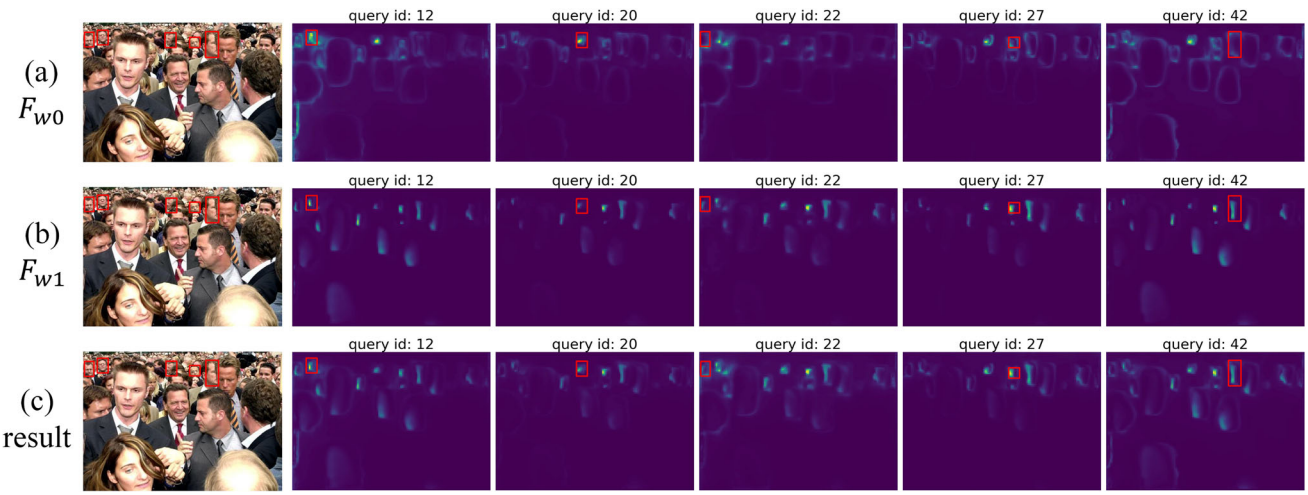

Fig. 5 Attention weight map of the cross-attention output. The first and second rows show the attention weight maps generated by the query vectors corresponding to the two sets of re-sampling points after performing cross-attention, and the third row shows the weight maps generated by the attention fusion method

average recall. The performance indicators and their meanings are shown in Table 1.

4.4 Analysis of experiment result

Table 2 shows the training results of the proposed FCDS-DETR model and the recently studied DETR variant model on the improved FDDB data set. As observed in Table 2, DETR performs poorly on the improved FDDB dataset and has the lowest AP and AR_{100} relative to the DETR variant. FCDS-DETR significantly improves the detection performance of DETR: +6.8 AP, +2.2 $AP_{0.5}$, and +6.0 $AP_{0.75}$ for 100 epochs, respectively. For the medium- and small-scale object detection, FCDS-DETR showed substantial improvements relative to DETR with +13.1 AP_M and +12.5 AP_S, respectively. The black bolded rows in the table indicate the experimental results of our proposed FCDS-DETR model and its application at multiple scales. The FCDS-DETR model also has significant performance advantages compared to the baseline model SAM–DETR model, with +1.4 AP, +1.1 $AP_{0.5}$, and +3.0 $AP_{0.75}$, respectively. The most significant improvements are found on AP_M and AP_S with +2.7 and +5.7, respectively. This demonstrates the effectiveness of the feature correction module and the double sampling mechanism in Sect. 3.3 feature correction module and Sect. 3.4 double sampling mechanism in improving the detector's performance. The performance of FCDS-DETR is even better than all the variants of DETR in the table. In addition, we can find that FCDS-DETR and the baseline model are similar in GFLOPs metrics, with FCDS-DETR increasing GFLOPs by a small amount (+7%). Figure 6 shows the detection results of FCDS-DETR with the baseline model SAM–DETR model on the improved FDDB dataset. Both models use ResNet-50 as the feature extraction network for 100 epochs. One of the (a) shows the original image, (b) shows the SAM–DETR model detection results, and (c) shows the FCDS-DETR detection results.

Meanwhile, we also conducted experiments on the COCO dataset, and the results are shown in Table 3. It can be observed that the convergence speed of FCDS-DETR at 12 epochs is not reduced compared to the baseline model but is improved considerably to some extent, which is a surprise to us. We posit that this improvement may be attributed to the enhancing effect of the feature correction module on the model's convergence during the early stages of training. FCDS-DETR improved by +5.3 AP, +6.0 $AP_{0.5}$, and +6.0 $AP_{0.75}$, respectively, compared to the original DETR after 50 epochs of training. It still performs outstandingly under 50 epochs of training, obtaining 39.0 AP, a +0.7 AP improvement over the baseline model. With the addition of multi-scale, the detection accuracy of FCDS-DETR is further improved, and 39.6 AP is obtained. Thanks to the high-quality sampling points and double sampling mechanism of FCDS-DETR, the model can perceive the position of the object to be detected more sharply during the training process. The convergence curves of each comparison model trained for 50 epochs on the COCO dataset are shown in Fig. 7. A large number of experiments fully demonstrate the effectiveness of our method. Figure 8 shows the detection results of FCDS-DETR with the baseline model SAM–DETR model on the COCO dataset. Both models use ResNet-50 as the feature extraction network for 100 epochs. One of the (a) shows the original image, (b) shows the SAM–DETR model detection results, and (c) shows the FCDS-DETR detection results.

In addition, it is worth noting that the FCDS-DETR model and the SMCA-DETR [21] model in Table 2 achieve similar scores on the AP metrics after 100 epochs of training, 76.4 and 76.5, respectively. We believe that the reason for the small difference in the performance of the two models is affected by the size of the dataset and the complexity of the scenes in the images. The improved FDDB dataset has 2.1k training images, the dataset size is small, and the complexity of the scene in which the target exists is low. After 100 epochs of training, both models can achieve better detection results. However, in Table 3, the performance gap between the models starts to appear when the two models undergo the same training epochs on the COCO dataset (118k training images) with complex scenes. FCDS-DETR achieves 39.0 AP after 100 epochs of training on the COCO dataset, which is an improvement of +0.6 over the SMCA-DETR model.

Table 1 Evaluation indicators

Index	Description
AP↑	Average precision at IoU = 0.50: 0.05: 0.95
AP_x ↑	Average precision at IoU=x
AP_L ↑	Average precision for large objects
AP_M ↑	Average precision for medium objects
AP_S ↑	Average precision for small objects
AR_{100} ↑	Average recall rate given 100 detections per image
AR_L ↑	Average recall rate for large objects
AR_M ↑	Average recall rate for medium objects
AR_S ↑	Average recall rate for small objects

↑ means bigger is better,and ↓ means smaller is better

Table 2 Comparison experiments—improved FDDB dataset

Model	Multi-scale	Epochs	Params(M)	GFLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR_{100}	AR_S	AR_M	AR_L
DETR-R50 [17]		100	41	86	69.6	91.0	80.7	3.0	56.7	82.4	76.2	18.2	67.5	87.4
Deformable DETR-R50 [19]		100	34	78	74.3	91.5	82.0	10.4	66.4	83.7	78.2	19.8	73.0	90.5
Conditional DETR-R50 [20]		100	43	195	72.8	92.8	83.3	7.7	63.1	84.0	77.8	17.4	70.7	88.1
SMCA-DETR-R50 [21]		100	42	86	76.5	93.1	86.9	15.8	69.5	86.3	81.1	22.0	75.9	89.8
SAM–DETR model [23]		100	58	100	75.0	92.1	83.7	9.8	67.1	85.5	79.9	79.3	73.2	89.8
FCDS-DETR-R50(Ours)		**100**	**64**	**107**	**76.4**	**93.2**	**86.7**	**15.5**	**69.8**	**86.2**	**81.3**	**21.1**	**75.0**	**91.1**
FCDS-DETR-R50*(Ours)	√	**100**	**61**	**186**	**77.7**	**93.2**	**88.0**	**20.8**	**71.2**	**86.5**	**82.3**	**27.9**	**77.0**	**90.8**

* Denotes the addition of multi-scale to the encoder, that is, the encoder in deformable DETR replaces the conventional transformer encoder

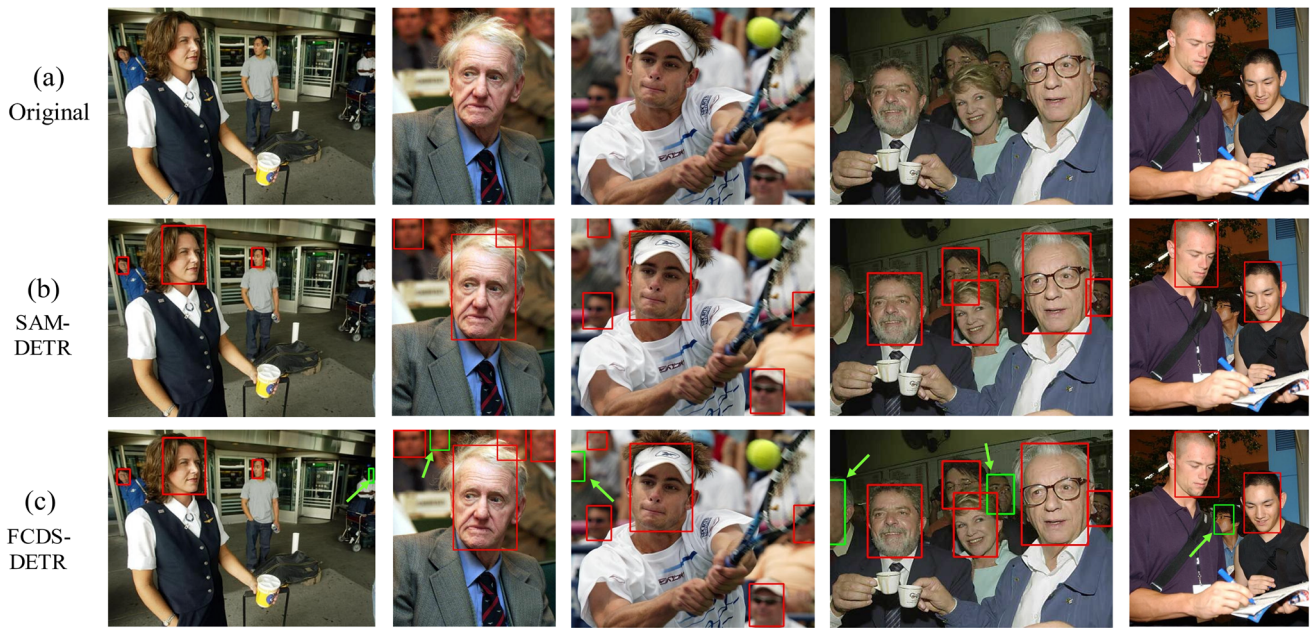

Fig. 6 On the improved FDDB dataset, our FCDS-DETR is more sensitive to the target objects, so the detection results are more accurate than the baseline model SAM–DETR model

4.5 Ablation analysis

To validate the role and contribution of the various components proposed in this paper for FCDS-DETR, we conducted an ablation study to assess the importance of the proposed feature correction module and the double sampling mechanism. Additionally, we compared the results with the baseline SAM–DETR model.

We used ResNet-50 as the feature extraction network for SAM–DETR model and FCDS-DETR. To eliminate variability, we performed 24 epochs of the baseline and FCDS-DETR models with different components added. The initial learning rate was 1×10^{-5}, the learning rate decreased to 1/10 of the original value after 16 epochs, and the number of object queries was set to 100. The experimental results are shown in Table 4. SAM–DETR model can achieve an AP of 34.4 after 24 epochs.

Table 3 Comparison experiment—COCO dataset

Model	Multi-scale	Epochs	Params(M)	GFLOPs	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR_{100}	AR_S	AR_M	AR_L
DETR-R50 [17]		12	41	86	16.8	33.1	15.4	4.3	16.3	28.8	36.4	10.5	38.3	61.5
Deformable DETR-R50 [19]		12	34	78	24.2	43.1	24.4	8.7	27.6	37.0	41.7	15.9	46.6	65.0
Conditional DETR-R50 [20]		12	43	195	24.4	43.3	24.4	8.3	26.6	38.8	42.6	15.7	46.9	67.9
SMCA-DETR-R50 [21]		12	42	86	24.4	44.2	23.8	9.3	27.2	36.8	42.9	16.9	47.6	67.9
SAM–DETR model [23]		12	58	100	28.0	48.5	27.9	9.9	30.7	45.0	44.5	16.9	49.2	71.5
FCDS-DETR-R50(Ours)		**12**	**64**	**107**	**29.3**	**50.0**	**29.3**	**10.9**	**32.3**	**46.7**	**45.3**	**19.0**	**49.8**	**70.9**
FCDS-DETR-R50*(Ours)	√	**12**	**61**	**186**	**36.2**	**54.6**	**38.2**	**19.5**	**39.7**	**50.5**	**53.5**	**29.6**	**57.4**	**75.9**
DETR-R50 [17]		100	41	86	33.7	55.0	34.2	13.4	36.2	52.5	50.7	22.8	55.5	76.6
Deformable DETR-R50 [19]		100	34	78	37.2	58.0	39.4	17.9	41.0	53.7	53.6	27.6	58.8	77.4
Conditional DETR-R50 [20]		100	43	195	38.3	59.4	40.2	17.7	41.6	56.8	54.7	27.1	60.1	79.7
SMCA-DETR-R50 [21]		100	42	86	38.4	59.5	40.6	18.3	41.5	57.0	54.8	28.8	60.1	79.8
SAM–DETR model [23]		100	58	100	38.3	60.2	39.9	17.7	41.8	57.9	53.8	26.9	59.5	78.9
FCDS-DETR-R50(Ours)		**100**	**64**	**107**	**39.0**	**61.0**	**40.2**	**18.1**	**42.3**	**59.6**	**54.4**	**28.0**	**60.1**	**79.6**
FCDS-DETR-R50*(Ours)	√	**100**	**61**	**186**	**39.6**	**61.3**	**41.1**	**19.8**	**43.2**	**60.4**	**54.9**	**30.3**	**61.1**	**79.9**

*Denotes the addition of multi-scale to the encoder, that is, the encoder in deformable DETR replaces the conventional transformer encoder

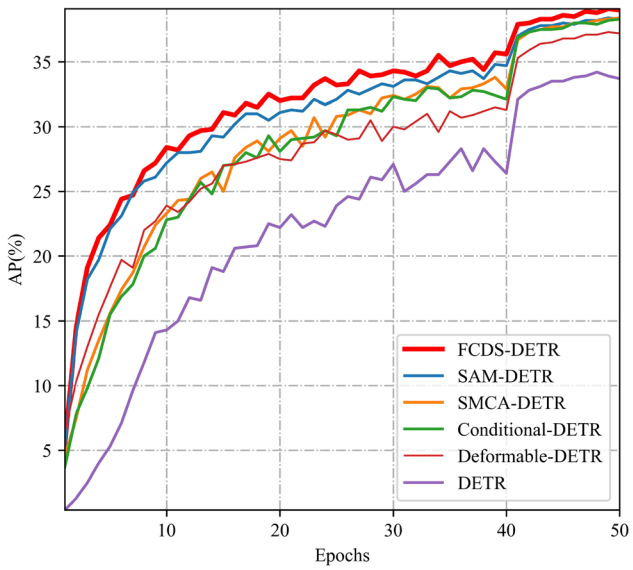

Fig. 7 Convergence curves of FCDS-DETR with other DETR variants trained on the COCO dataset for 50 epochs. Compared with the original DETR, FCDS-DETR significantly improved AP while outperforming other DETR variants

4.5.1 Effectiveness of feature correction module

As shown in Table 4, after ensuring the same learning rate, number of epochs, and training schedule as the baseline model, we obtain the results by adding different feature correction methods to the semantic alignment module. It can be observed that the improved models after adding the feature correction module all have different degrees of improvement in accuracy compared with the baseline model. The addition of SENet [58] improves the model accuracy by +0.5 AP, +0.7 $AP_{0.5}$, and +0.7 $AP_{0.75}$, respectively. Adding CAM [55] improves the model accuracy by +1.0 AP, +1.1 $AP_{0.5}$, and +1.4 $AP_{0.75}$, respectively. The addition of CBAM [59] improves the model accuracy by +0.1 AP, +0.6 $AP_{0.5}$, and +0.2 $AP_{0.75}$. The improved model corresponding to CBAM has limited improvement to the model when the number of epochs is small because it needs to learn two dimensions of channel and space. The results show that adding the feature correction module improves the detection accuracy of the

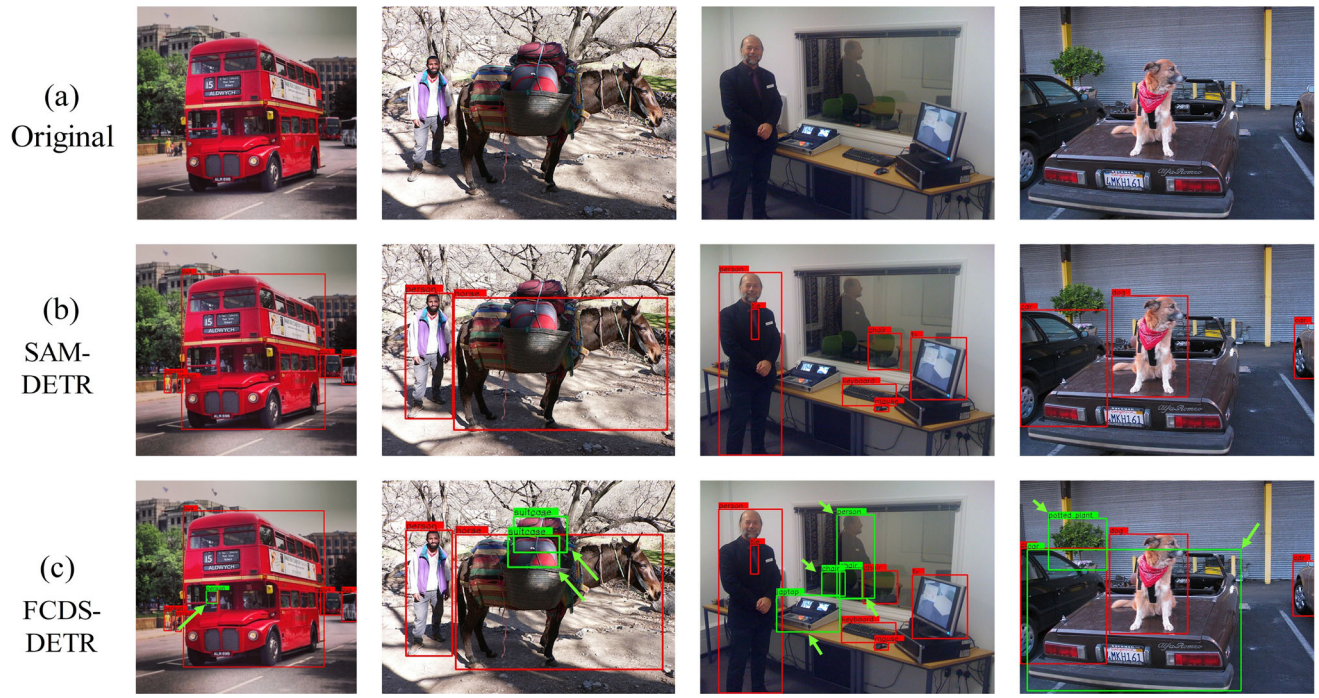

Fig. 8 Detection results of FCDS-DETR and baseline model on COCO dataset. FCDS-DETR can detect small- and medium-sized objects in the image, and the positioning of the bounding box is more accurate

Table 4 Ablation experiments

Row	FCDS	SENet	CAM	CBAM	Double Sampling	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L	AR_{100}
1						34.4	55.9	35.2	15.3	37.9	53.0	49.1
2	√	√				34.9	56.6	35.9	15.6	38.0	54.3	51.3
3	√		√			35.4	57.0	36.6	15.9	38.5	55.0	51.4
4	√			√		34.5	56.5	35.4	14.4	37.8	53.5	50.4
5	√				√	35.3	57.0	36.8	16.1	38.2	54.8	51.5
6	√		√		√	35.9	57.2	37.3	16.5	39.1	55.7	51.7

model while maintaining the advantage of the baseline model in terms of convergence speed.

4.5.2 Effectiveness of the double sampling mechanism

As shown in Table 4, we added a double sampling mechanism to the semantic alignment module for double sampling within the region containing potential objects. We used the attention fusion method described in Sect. 3.6 to accommodate the attention map fusion problem due to doubling the number of sample points for double sampling. A comparison with the baseline model shows that the improved model with the inclusion of the double sampling mechanism improves the detection accuracy by +0.9% AP for the same number of epochs. This is a very impressive result and strongly supports our view that the double sampling mechanism improves the sensitivity of the model to target objects.

4.6 Visualization

Figure 9 visualizes the bounding boxes predicted by FCDS-DETR and the corresponding key points searched by applying the double sampling mechanism. Among them, the first set of sampling points is marked with green, and the second set of sampling points is marked with blue. Meanwhile, to show the advantage of FCDS-DETR in extracting object features, we also visualize the weight maps generated by the FCDS-DETR and SAM–DETR models after 50 epochs on the COCO dataset. Figure 10 visualizes the attention fusion method by visualizing the

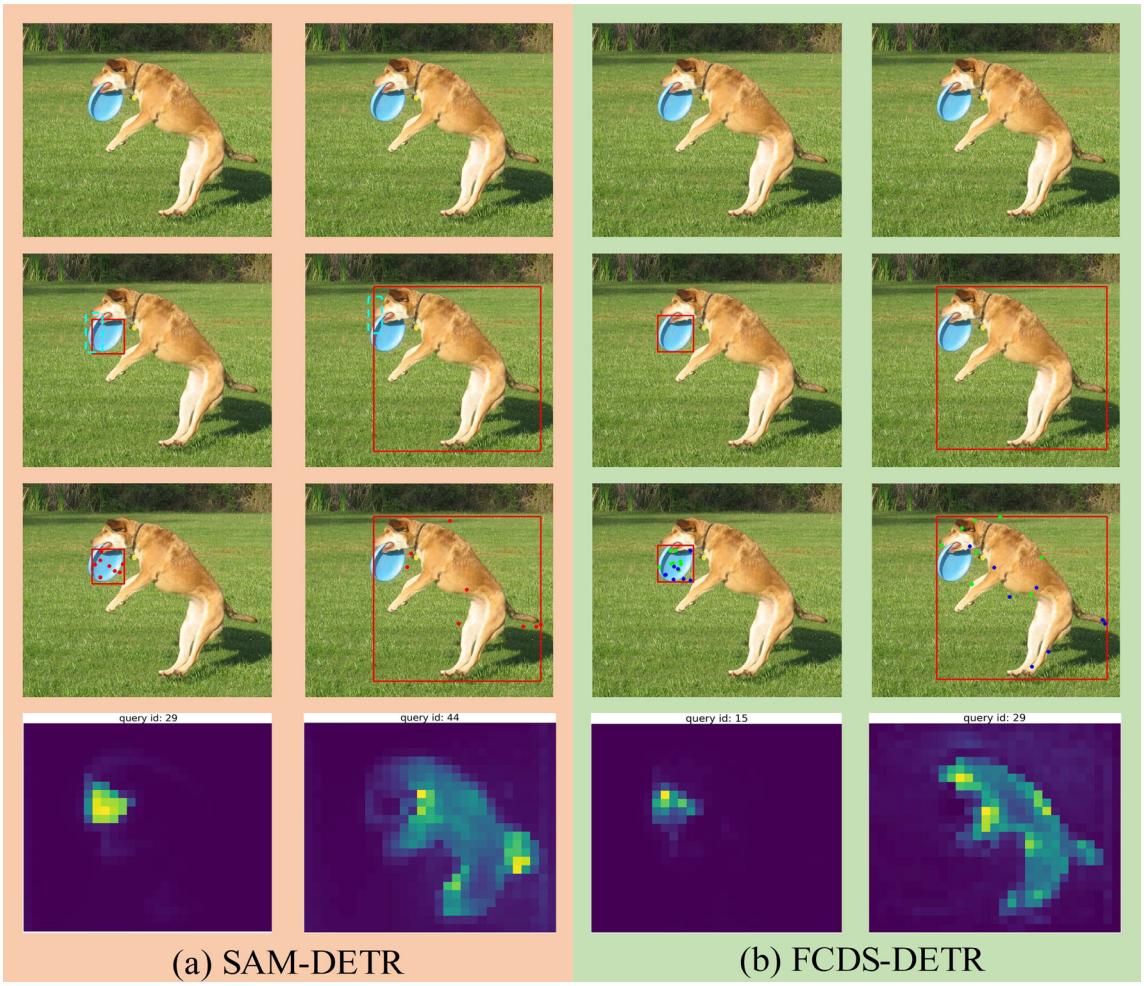

Fig. 9 Detection results and weight maps obtained on the COCO dataset. The first row shows the original image to be detected. The second row visualizes the location of the bounding boxes. The third row visualizes the localization of the sampling points in the bounding boxes. The fourth row visualizes the final generated weight maps. Our FCDS-DETR can locate the edge and end of the object more accurately, making the bounding boxes localization more accurate

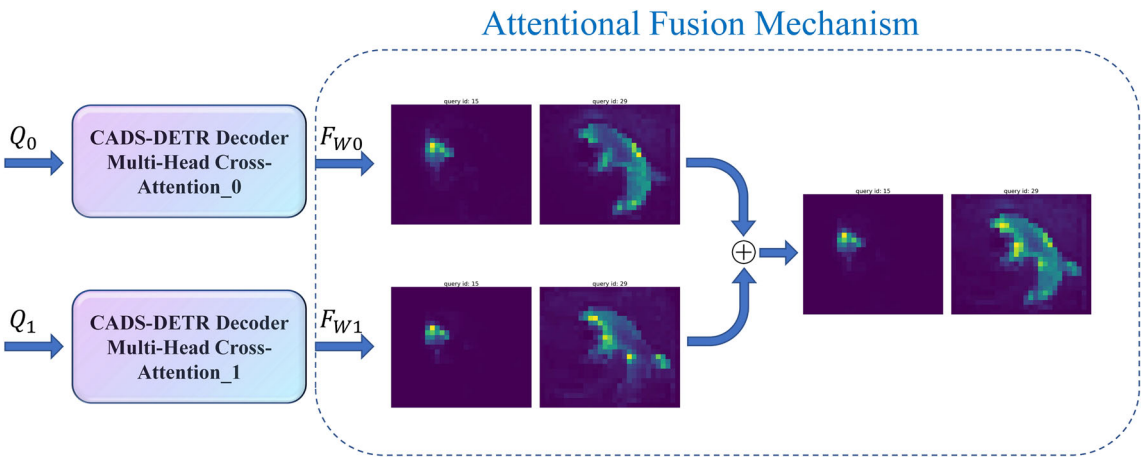

Fig. 10 Attentional fusion method

weight maps generated by the multi-head cross-attention module.

It can be observed that double sampling on feature maps to which the feature correction module is applied allows more sampling points to be accurately located at the edges or ends of the objects to be detected. These sampling areas with important features play a crucial role in the subsequent object localization and recognition. Meanwhile, the comparison of the attention weight maps generated by the cross-attention modules of the two models shows that the weight maps obtained using the attention fusion method can clearly separate the object to be measured from the background. Therefore, this model can sensitively perceive and locate the object in the image and improve the accuracy of object detection. In contrast, the original SAM–DETR model has more scattered and sparse sampling points when a single re-sampling is performed, which cannot locate the edges and ends of the object well. It can be observed from the generated weight maps that SAM–DETR model is also less sensitive to the objects. Such results are consistent with our analysis above that fewer sampling points and a blurred attention map are the main reasons for SAM–DETR model's low detection accuracy.

5 Conclusion

In this paper, we discuss the reasons for the unsatisfactory detection accuracy of SAM–DETR model when performing object detection, i.e., fewer sampling points and blurred attention. We propose FCDS-DETR to solve the above problems and obtain better performance. The core idea of FCDS-DETR is to improve the accuracy and number of sampling points localization by adding a feature correction module and double sampling mechanism, thus improving the recognizability of the attention map output by the model. We demonstrate the effectiveness of the model through a large number of experiments.

The limitations of our proposed FCDS-DETR model are shown in two aspects. On the one hand, the output of the cross-attention module may superimpose the background noise existing in the two attention weight maps when performing feature fusion, which may adversely affect the detection performance of the model. On the other hand, the Semantics Aligner module does not implement the combination with other improved DETR model methods for the time being, which affects the further improvement of model detection performance. To address the above limitations, we will continue to explore more effective noise reduction algorithms for application in the attention fusion process in the future. At the same time, we will continue to investigate the fusion between FCDS-DETR and other excellent improvements to achieve more excellent detection performance.

Data availability All data generated or analyzed during this study are included in this published article. Derived data supporting the findings of this study are available from the corresponding author on request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. Int J Comput Vis. 128:261–318
2. Bodla N, Singh B, Chellappa R, Davis LS (2017) Soft-NMS–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision, pp 5561–5569
3. Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6154–6162
4. Fan Q, Zhuo W, Tang CK, Tai YW (2020) Few-shot object detection with attention-rpn and multi-relation detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4013–4022
5. Hu H, Gu J, Zhang Z, Dai J, Wei Y (2018) Relation networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3588–3597
6. Lyu P, Liao M, Yao C, Wu W, Bai X (2018) Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of the European conference on computer vision (ECCV), pp 67–83
7. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
8. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28
9. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: Computer vision–ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, pp 21–37
10. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
11. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
12. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
13. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, Ke Z, Li Q, Cheng M, Nie W, et al. (2022) Yolov6: a single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976
14. Wang CY, Bochkovskiy A, Liao HYM (2022) Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696

15. Ge Z, Liu S, Wang F, Li Z, Sun J (2021) Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
17. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, Springer, pp 213–229
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30
19. Zhu X, Su W, Lu L, Li B, Wang X, Dai J (2020) Deformable DETR: deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159
20. 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
21. Gao P, Zheng M, Wang X, Dai J, Li H (2021) Fast convergence of DETR with spatially modulated co-attention. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3621–3630
22. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum HY (2022a) Dino: DETR with improved DeNoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605
23. Zhang G, Luo Z, Yu Y, Cui K, Lu S (2022b) Accelerating DETR convergence via semantic-aligned matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 949–958
24. Jain V, Learned-Miller E (2010) Fddb: A benchmark for face detection in unconstrained settings. Tech. rep, UMass Amherst technical report
25. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: Computer Vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, pp 740–755
26. Hamano G, Imaizumi S, Kiya H (2023) Effects of jpeg compression on vision transformer image classification for encryption-then-compression images. Sensors 23(7):3400
27. Roy SK, Deria A, Hong D, Rasti B, Plaza A, Chanussot J (2023) Multimodal fusion transformer for remote sensing image classification. IEEE Trans Geosci Remote Sens
28. Zheng Y, Gindra RH, Green EJ, Burks EJ, Betke M, Beane JE, Kolachalama VB (2022) A graph-transformer for whole slide image classification. IEEE Trans Med Imaging 41(11):3003–3015
29. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y (2021) Transformer in transformer. Adv Neural Inf Process Syst 34:15908–15919
30. Xu X, Xu N (2022) Hierarchical image generation via transformer-based sequential patch selection. Proc AAAI Conf Artif Intell 36:2938–2945
31. Esser P, Rombach R, Ommer B (2021) Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12873–12883
32. Zhang B, Gu S, Zhang B, Bao J, Chen D, Wen F, Wang Y, Guo B (2022) Styleswin: Transformer-based GAN for high-resolution image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11304–11314
33. Chang H, Zhang H, Jiang L, Liu C, Freeman WT (2022) Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11315–11325
34. Plizzari C, Cannici M, Matteucci M (2021a) Spatial temporal transformer network for skeleton-based action recognition. In: Pattern recognition. ICPR international workshops and challenges: virtual event, January 10–15, 2021, Proceedings, Part III, Springer, pp 694–701
35. Plizzari C, Cannici M, Matteucci M (2021) Skeleton-based action recognition via spatial and temporal transformer networks. Comput Vis Image Underst 208:103219
36. Li X, Hou Y, Wang P, Gao Z, Xu M, Li W (2021) Trear: transformer-based RGB-D egocentric action recognition. IEEE Trans Cognit Dev Syst 14(1):246–252
37. Yu S, Wang M, Pang S, Song L, Qiao S (2022) Intelligent fault diagnosis and visual interpretability of rotating machinery based on residual neural network. Measurement 196:111228
38. Yu S, Wang M, Pang S, Song L, Zhai X, Zhao Y (2023) Tdmsae: a transferable decoupling multi-scale autoencoder for mechanical fault diagnosis. Mech Syst Signal Process 185:109789
39. Zhao G, Lin J, Zhang Z, Ren X, Sun X (2019) Sparse transformer: concentrated attention through explicit selection
40. Wang S, Li BZ, Khabsa M, Fang H, Ma H (2020) Linformer: self-attention with linear complexity. arXiv preprint arXiv:2006.04768
41. Messina N, Falchi F, Esuli A, Amato G (2021) Transformer reasoning network for image-text matching and retrieval. In: 2020 25th international conference on pattern recognition (ICPR), IEEE, pp 5222–5229
42. Mueller J, Thyagarajan A (2016) Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the AAAI conference on artificial intelligence, vol 30
43. Chen Q, Zhu X, Ling Z, Wei S, Jiang H, Inkpen D (2016) Enhanced LSTM for natural language inference. arXiv preprint arXiv:1609.06038
44. Chen H, Luo Z, Zhou L, Tian Y, Zhen M, Fang T, Mckinnon D, Tsin Y, Quan L (2022) Aspanformer: detector-free image matching with adaptive span transformer. In: European conference on computer vision, Springer, pp 20–36
45. Chen J, Chen X, Chen S, Liu Y, Rao Y, Yang Y, Wang H, Wu D (2023) Shape-former: Bridging CNN and transformer via ShapeConv for multimodal image matching. Inf. Fusion 91:445–457
46. Liao S, Shao L (2021) Transmatcher: deep image matching through transformers for generalizable person re-identification. Adv Neural Inf Process Syst 34:1992–2003
47. Su W, Wang Y, Li K, Gao P, Qiao Y (2023) Hybrid token transformer for deep face recognition. Pattern Recogn 139:109443
48. Li X, Du J, Yang J, Li S (2022) When mobilenetv2 meets transformer: a balanced sheep face recognition model. Agriculture 12(8):1126
49. Luo M, Wu H, Huang H, He W, He R (2022) Memory-modulated transformer network for heterogeneous face recognition. IEEE Trans Inf Forensics Secur 17:2095–2109
50. Chopra S, Hadsell R, LeCun Y (2005) Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), IEEE, vol 1, pp 539–546
51. Koch G, Zemel R, Salakhutdinov R, et al. (2015) Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, Lille, vol 2
52. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
53. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759–8768

54. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
55. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2921–2929
56. 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
57. Loshchilov I, Hutter F (2017) Fixing weight decay regularization in Adam
58. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
59. Woo S, Park J, Lee JY, Kweon IS (2018) Cbam: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp 3–19