IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 1, 2021, accepted September 18, 2021, date of publication September 29, 2021, date of current version October 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3116324

Transformed Dynamic Feature Pyramid for Small Object Detection

HONG LIANG, (Member, IEEE), YING YANG, (Member, IEEE), QIAN ZHANG, (Member, IEEE), LINXIA FENG, (Member, IEEE), JIE REN, (Member, IEEE), AND QIYAO LIANG, (Member, IEEE)
College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, Shandong 266580, China

Corresponding author: Ying Yang (18336367235@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673396, and in part by the Natural Science Foundation of Shandong Province under Grant ZR2020MF136.

ABSTRACT The low resolution and less feature information of small targets make it difficult to recognize and locate, which greatly hinders the improvement of object detection accuracy. In this paper, an object detection model (TDFP) based on CNN and transformer was established, which combines local and global context to establish the connection between features. In the proposed transformed dynamic feature pyramid network, a transformer module was designed to dynamically transform and fuse the multi-scale features generated by the backbone to generate a transformed feature pyramid with richer multi-scale features and context information. In this transformation process, gate block is used to dynamically select single-scale transformation or cross-scale transformation to achieve an optimal style of transformation and fusion of multi-scale features. The experimental results show that the model improves the small targets detection accuracy based on CNN and transformer. Based on the backbone ResNeXt-101, TDFP achieves 46.2% AP and 26.3% AP_S on MS COCO, and takes the amount of computation as a loss constraint to achieve a better balance between detection accuracy and computational complexity.

INDEX TERMS Local and global context information, transformer module, transformed feature pyramid, single-scale transformation, cross-scale transformation.

I. INTRODUCTION

In recent decades, object detection methods based on convolutional neural networks (CNN [1]–[4]) have made great achievements. However, the low detection accuracy of small targets is a difficult problem in object detection, which hinders the further improvement of object detection accuracy. Therefore, the researchers proposed various solutions, such as better multi-scale feature fusion methods [6]–[10], richer context information [11]–[14], appropriate training method [15], denser anchor sampling and matching strategies [16]–[20]. Most of these methods depend on CNN and the preset of anchor boxes. However, the long-distance dependence between objects in images is very important in visual tasks. For image data, CNN can only capture the long-distance dependence between targets by the large receptive field generated by repeated convolution operations [21], [22], which leads to a complex calculation.

In recent years, it has been found that self-attention [23] and non-local [24] operations can capture the interaction between targets. Compared with CNN, self-attention in the transformer can mining long-distance dependence between targets and is not limited by the inductive bias of local interaction, and has strong expression ability. Therefore, the transformer is extended to various specific tasks in computer vision, such as classification [25]–[27], object detection [28]–[33] and segmentation task [34], [35], etc., and obtains global information through self-attention. But compared with CNN-based two-stage detectors and one-stage detectors, transformer-based methods have a little disadvantage in detection accuracy. Convolution has translation invariance and local sensitivity, but it lacks the overall perception and macro understanding of the image. The transformer can be used in a convolution network to learn the global features of images. However, for high-resolution input, the self-attention layer is more computational, so it is suitable for smaller spatial dimension input. Therefore, it is worth further research to optimize the network based on the advantages of CNN and transformer.

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

In this study, we use the convolution method to learn the visual part with rich local context efficiently, and then use the transformer method to learn global context information. Based on CNN and transformer, a new detection model (Transformed Dynamic Feature Pyramid, TDFP) is proposed, whose core is a transformed dynamic feature pyramid network. In this network, a transformer module is designed. After the backbone generating multi-scale features, the better multi-scale feature fusion mode is realized by dynamically selecting the cross-scale transformation and single-scale transformation via gate block and capturing the local and global context information to establish the relationship between the targets. And the transformed feature pyramid with richer multiscale features and context information is generated to alleviate the small targets problem. In addition, to reduce the calculation, we take the calculation as the constraint loss to achieve the optimal balance between the detection accuracy and the calculation.

The detection method proposed in this paper has the following advantages:

(1) Compared with the previous CNN-based and transformer-based detection methods, richer multi-scale features and context information can be obtained.

(2) Through dynamic feature transformation and fusion, our model can get richer multi-scale features and context information, and the detection accuracy of small targets based on CNN and transformer has been improved greatly.

(3) The computation is used as the loss constraint to achieve the optimal balance of detection accuracy and computational complexity.

II. RELATED WORK

A. CNN-BASED DETECTORS

Two-stage detectors based on CNN, RCNN [36] and its variants [18], [37], [38] solved the problems of traditional detectors with hand-designed features, such as many steps, high time complexity, window redundancy, poor detection accuracy [39], and achieved high detection accuracy, but lack of real-time.

YOLO [40] and its variants [41]–[43], SSD [44] and its variants [16], [45], [46] avoid the use of RPN and realize real end-to-end detection. Some networks can achieve real-time detection while maintaining high detection accuracy, but the detection accuracy of most two-stage detectors is lower than that of two-stage detectors. Scaled-YOLOv4 [41] proposed a network scaling approach that modifies not only the depth, width, resolution but also the structure of the network. YOLOr [42] proposed a unified network to encode implicit knowledge and explicit knowledge together, which can generate a unified representation to simultaneously serve various tasks and benefit the performance of all tasks. RetinaNet [47] proposed focal loss to solve the problem of class imbalance to improve detection accuracy. Lu *et al.* [48] proposed a novel and effective framework, MimicDet, which has a shared backbone for one-stage and two-stage detectors, then it branches into two heads which are well designed to have compatible features for mimicking, to train a detector by directly imitating two-stage functions. However, most of the above detectors rely on manually set anchor boxes to achieve the detection task. The setting of the anchor involves many parameters and has complex computation. The final performance of the model is sensitive to the anchor boxes, so the robustness of the model is poor.

In recent years, the center-based methods [49]–[51] and the keypoints-based methods [52]–[54] have eliminated the use of anchors, but the detection accuracy is low. ATSS [55] showed that the essential difference between anchor-based detectors and anchor-free detectors is actually how to define positive and negative training samples, and proposed an adaptive training sample selection approach to automatically selects positive and negative training samples according to the statistical characteristics of the targets. which can improve the performance of detectors.

Recently, [3] proposed SpineNet, a backbone with scale-permuted intermediate features and cross-scale connections that was learned on an object detection task by Neural Architecture Search(NAS). The learned scale-permuted model outperforms ResNet-50-FPN by (+2.9% AP) in the object detection task. The efficiency can be further improved (−10%FLOPs) by adding search options to adjust the scale and type of each candidate feature block. Cascade RCNN-RS [37] provided simple scaling strategies to generate a family of models that form two Pareto curves, named RetinaNet-RS and Cascade RCNN-RS. These simple rescaled detectors explore the speed-accuracy trade-off between the one-stage RetinaNet detectors and two-stage RCNN detectors. They identified the key architectural changes, training methods and inference methods that significantly improve object detection and instance segmentation systems in speed and accuracy. Zhou *et al.* [56] developed a probabilistic interpretation of two-stage object detection, which motivates a number of common empirical training practices. They presented a simple modification of standard two-stage detector training by optimizing a lower bound to a joint probabilistic objective over both stages. The resulting detectors are faster and more accurate than both their one- and two-stage precursors.

B. TRANSFORMER-BASED DETECTORS

In the research of applying transformer to computer vision tasks, Cordonnier *et al.* [57] proposed that the self-attention layer can also achieve the same effect as the convolution layer, while reducing the computational complexity, and can replace the convolution layer. Transformer-based methods can be divided into [57]: (1) vanilla transformer replaces convolutional neural network to achieve visual tasks [26], [27]. Beal *et al.* [61] used Vit [26] as the backbone network, combined with a prediction head to achieve the final detection, the detection effect of large targets is good, but with poor detection effect of small targets. Therefore, the use of vanilla transformer still needs further research.(2) Combine

transformer with CNN. Detr [28] for the first time combines the transformer with CNN for object detection and achieves the SOTA performance, which simplifies the detection pipeline, regards the target detection as an unordered set prediction problem, and compulsorily realizes the unique prediction through binary matching. However, the binary matching between transformer decoder and Hungarian loss is unstable, which leads to slow convergence speed and poor detection effect of small targets. The FPT proposed by Tong *et al.* [39] is to apply the idea of transformer to the transformation of feature pyramid [60]. Three specially designed transformers are used to transform any feature pyramid into another feature pyramid of the same size but with a richer context in a top-down and bottom-up interactive way, so as to alleviate the small target problem.

To solve the convergence problem of Detr, deformable Detr [30] was proposed to use a deformable attention module instead of the original multi-head attention to focus on a small group of key positions around the reference point. Sun *et al.* [32] proposed the encoder-only version of Detr, designed a new binary matching scheme to achieve more stable training and faster convergence, and proposed two ensemble prediction models TSP-FCOS and TSP-RCNN based on transformer, which have better performance than the original Detr model, and greatly improved the detection accuracy and training convergence.

For the high computational complexity of Detr, Srinivas *et al.* [33] proposed an adaptive clustering transformer (ACT) to reduce the computational cost of pre-trained Detr without any training process. LeCun *et al.* [22] only uses global self-attention to replace the last three bottlenecks of ResNet [1], which significantly improves the baseline in instance segmentation and object detection, while reducing the cost of parameters and minimizing latency.

Recently, [62] constructed a hierarchical transformer and introduced the idea of the locality to calculate the self-attention [23] in the non-overlapping window area, which greatly reduced the computational complexity and improved the detection accuracy. Yang *et al.* [63] presented focal self-attention, a new mechanism that incorporates both fine-grained local and coarse-grained global interactions. They also proposed a new variant of Vision Transformer models with focal self-attention, called Focal Transformer, which achieves superior performance over the state-of-the-art vision Transformers [27] on a range of public image classification and object detection benchmarks. Meanwhile, Dai *et al.* [64] presented a novel dynamic head framework to unify object detection heads with attention. The proposed approach significantly improves the representation ability of object detection heads without any computational overhead by coherently combining multiple self-attention mechanisms between feature levels for scale-awareness, among spatial locations for spatial awareness, and within output channels for task-awareness.

C. MULTI-SCALE FEATURE FUSION

For the fusion of multi-scale features, the most direct method is to add multi-scale features [60], [65]. FPN [60] is the first time to propose a feature pyramid with top-down and horizontal connections to solve multi-scale problems, especially the small target problem. PANet [65] adds a bottom-up path to FPN [6]–[8]. Different multi-scale feature fusion methods are used to generate a better feature pyramid [6]–[8]. PFPNet [6] constructs the feature pyramid by widening the network width instead of increasing the network depth. AugFPN [7] considers the difference between different scale features and uses the adaptive feature fusion method to add multiscale features with weights. A new architecture of FPN reconfiguration [8] is proposed, which can aggregate task-oriented features in different spatial locations and scales.

Another method is to connect multi-scale features along the channel direction [10], [66]. In addition, some studies [23], [24], [67] consider the information interaction within the same scale features.

III. METHODOLOGY

A. OVERALL ARCHITECTURE

The network architecture proposed in this paper mainly includes three parts:

(1) The backbone. The pre-trained ResNet is used as the backbone to extract the multi-scale feature maps $\{a, b, c\}$ of the input image. The size of the input image is 800×1000. The size and lower sampling rate s corresponding to $\{a, b, c\}$ are $25 \times 32/32$, 50×68 / 16 and $100 \times 125/8$.

(2) The Transformed dynamic feature pyramid network. $\{a, b, c\}$ are transformed into a transformed feature pyramid with richer multi-scale features and context information to alleviate the small target problem via the designed transformer module. The details are shown in sections 3.2-3.4.

(3) Head network. At the top of the transformed feature pyramid, Fast RCNN [38], which is the head network, is used to implement the detection task. In order to enhance the generalization ability of the model and avoid overfitting, drop block [68] is applied to each output feature graph. The drop block size is 5 and the feature retention probability is 0.9.

B. TRANSFORMED DYNAMIC FEATURE PYRAMID NETWORK

Transformed dynamic feature pyramid network is the core part of the detection model proposed in this paper, as shown in Figure 2. It contains three parts:

(1) The multi-scale features $\{a, b, c\}$. They are generated by the backbone.

(2) Transformer module. It uses gate block to dynamically change the feature transformation and fusion methods to achieve better feature fusion.

(3) Transformed feature pyramid with local and global information. More abundant multi-scale features and context information are aggregated via the Transformer module.

1) TRANSFORMER MODULE

The transformer module consists of single-scale transformation, cross-scale transformation and gate block. There is a semantic gap between multi-scale features, so it is difficult to make a better fusion of features and establish the relationship between them if the horizontal connection and top-down fusion are carried out directly. At the same time, through multiple down sampling, features lose the underlying spatial location information, which is harmful to small target detection. Therefore, the idea of transformer is applied to the above multi-scale features, and the gate block is used to dynamically select the transformation and fusion methods of features, so as to capture the global context information of multi-scale features to establish the relationship between features and the feature pyramid with richer multi-scale feature information. The details of the transformation are described in Section 3.3.

To achieve better multi-scale feature fusion, gate block is used to select whether to carry out cross-scale transformation and fusion to obtain features b_t, so as to acquire the relationship between different scale features. In order to explain the process of transformation and fusion more clearly, the transformation process of one layer of features is described, taking the transformation and fusion of generated features as an example. The transformation between the same color features is the same scale transformation, and the transformation between different color features is the transformation of different sizes. The feature maps $\{a_1, b_1, c_1\}$ obtained by feature transformation has the same scale as the original features $\{a, b, c\}$. If only single-scale transformation is used, the features $\{a_1, b_1, c_1\}$ with the same size are obtained after one transformation. In order to retain the spatial position information of the original feature, the feature $\{a_1, b_1, c_1\}$ is connected with the feature b. If cross-scale transformation is used, feature $\{a, b, c\}$ carries out three transformations respectively, and three features with the same scale are added to get the features $\{a_1, b_1, c_1\}$. In order to simplify the network structure, the adding process between the three features is omitted, and the transformed features $\{a_1, b_1, c_1\}$ are obtained directly. Then, it is connected with the original feature b and the feature obtained by the branch of single scale transform in the direction of channel. Then the features obtained from the above connection are processed by a convolution of 1×1. The dimension of the feature is reduced to 256, and the transformed feature b_t is obtained by adding the feature with two times of a_t up sampling.

The above operations are performed on features $\{a, b, c\}$ respectively, and a top-down path is added between features $\{a_t, b_t, c_t\}$ to obtain the transformed feature pyramid $\{a_t, b_t, c_t\}$. Compared with the size of input image, the bottom-up feature size of $\{a_t, b_t, c_t\}$ is 8, 16 and 32 of down sampling rate respectively, and the feature of each layer has the same channel number of 256.

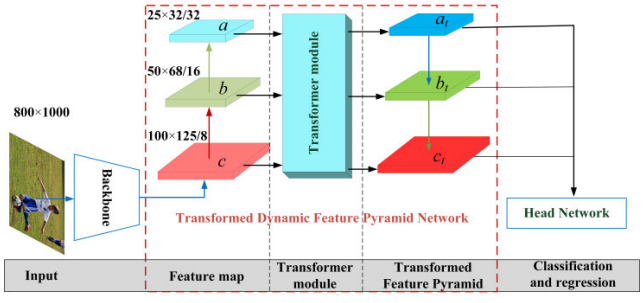


FIGURE 1. Overall network architecture.

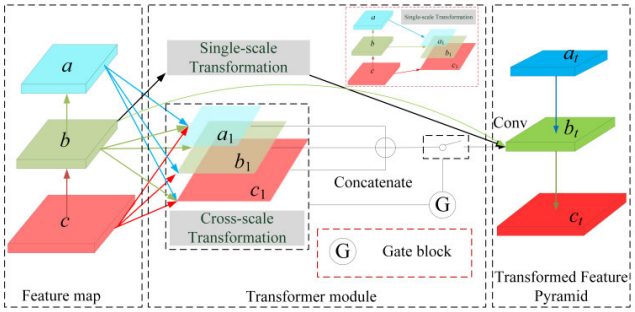


FIGURE 2. Transformed dynamic feature pyramid network.

2) SELF-ATTENTION

Self-attention is the core idea of transformer. The input of self-attention layer is a feature graph, and an updated feature map is obtained for the purpose of calculating the attention weight between each pair of features, each of which contains information about any other location in the same image. If each position in the feature map is a random variable, the similarity between any two positions is calculated. The value of each predicted pixel is enhanced or weakened according to the similarity between each predicted pixel and other pixels in the image. Similar pixels are used in training and prediction, and different pixels are ignored. Self attention layer can deal with the larger sense field than conventional convolution, so these models can obtain the dependence between the features with long-distance interval in space.

For self-attention, it is usually in the form of scaled-dot-product [23]. Given query matrix, key matrix and value matrix, the correlation between the two is first calculated by multiplying and dividing by scaling factor, and then the weighted sum of the result and value vector is finally output.

$$Attention(Q, K, V) = soft\max(QK^T \Big/ \sqrt{d_k}) \cdot V \tag{1}$$

$$Q = W_q X_{i,j}, \quad K = W_k X_{a,b}, \quad V = W_v X_{a,b} \tag{2}$$

where Q, K, V are transformer matrices. $X_{i,j}, X_{a,b}$ represent different input feature maps.

C. SINGLE-SCALE TRANSFORMATION AND CROSS-SCALE TRANSFORMATION

1) SINGLE-SCALE TRANSFORMATION

The process of single-scale transformation is shown in Figure 3, mainly considering the relationship between pixels in the same feature map. The multi-scale features $\{a, b, c\}$ are transformed by a single-scale transformation to get the features $\{a_1, b_1, c_1\}$, which is similar to the operation of self-attention layer. The specific change process is as follows.

This section takes the single-scale transformation process of spatial range feature map as an example (as shown in Figure 4). Given a pixel in the feature map, we first extract a region as the center, and the pixel position of the region is, which is the number of pixels. After a single head attention layer, the output of the pixel is as follows,

$$y_{i,j} = \sum_{a,b \in N_k(i,j)} soft\max_{a,b}(q_{i,j}^T k_{a,b}) \cdot v_{a,b} \tag{3}$$

where queries, keys and values are linear transformations of position pixels and adjacent pixels, which means that a number adjacent to the location is applied and then sum them. When local self-attention gathers spatial information on neighborhood similar to convolution, aggregation is accomplished by convex combination of value vectors with mixed weight, and the mixed weights are parameterized by content interaction. Repeat this calculation for each pixel to get the updated feature map, which has the same scale as the feature map.

In most cases, multiple attention heads are used to learn a variety of different representations of input. The principle of the method is to divide the pixel feature depth into groups. As described above, the attention of each group is calculated separately. Each head uses different transformations and then connects the output representation to obtain the final output.

2) CROSS-SCALE TRANSFORMATION

There is a semantic gap between multi-scale features. In order to better realize the feature interaction between multi-scale features, cross-scale transformation is used to calculate between two different scale features to get the transformed feature map. Firstly, the features are transformed by single-scale and cross-scale transformation respectively, and then the two transformed feature maps of the same scale are added to get the features. Take the cross-scale transformation of feature as an example. Given a feature map, the output feature graph and the feature graph have the same size. Euclidean distance is used as the similarity function to calculate the similarity.

$$F_{eud}(q_i, k_j) = -||q_i - k_j||^2 \tag{4}$$

where $q_i = f_q(\chi_i^b)$ and $k_j = f_q(\chi_j^a)$. χ_i^b is the i^{th} position of χ^b, χ_j^a is the j^{th} position of χ^a. q_i, k_j is divided into N parts, We get the process of cross-scale transformation as follows,

$$\begin{aligned} &Input : q_i, k_j, v_j, N \\ &Similarity : s_{i,j}^n = F_{eud}(q_{i,n}, k_{j,n}) \end{aligned}$$

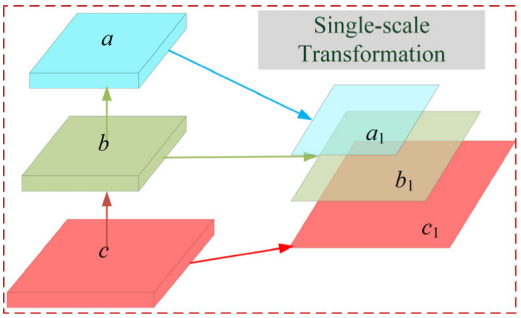


FIGURE 3. Single-scale transformation.

$$\begin{aligned} &Weight : w_{i,j} = softmax(s_{i,j}^n) \\ &Output : \tilde{\chi}_i^b = F_{mul}(w_{i,j}, v_j) \end{aligned} \tag{5}$$

where $v_j = f_v(\chi_j^a)$ is the similarity score of the part χ_j^a, and $s_{i,j}^n$ is the feature position of the middle transformation. F_{mul} is dot product When each pair has a closer distance, they will be given a greater weight. The cross-scale transformation of other scale features is the same.

D. GATE BLOCK

In this paper, we use gate block (as shown in Figure.5) to dynamically change the feature transformation and fusion methods to get better feature fusion. Single-scale transformation is a branch that must participate in the transformation, and cross-scale transformation is decided by gate block. If both branches participate in the transformation at the same time, the transformed features are added to generate each layer of the transformation feature pyramid. CNNGate block [67] is used as the gate block. CNNGate block includes an average pooling layer, two fully connected layers, a ReLU activation function and GumbelSoftmax [69]. The transformed features $\{a_1, b_1, c_1\}$ are passed through CNNGate block. Assume that the input features with the shape of (C, H, W) are first compressed by the average pooling operation, and the feature dimension is reduced to 1 / 4 of the original dimension. C,H,W are the number, height and width of feature channels. Then, two full join layers, a nonlinear activation function ReLU and a GumbelSoftmax function, are

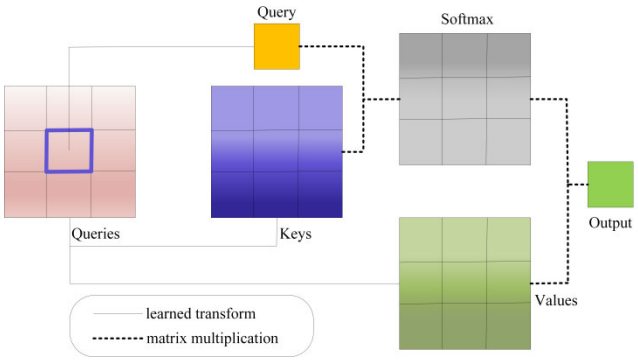


FIGURE 4. An example of a local attention layer in a $k = 3$ spatial range.

used to generate a one hot gate vector β_l for dynamic blocks.

$$\begin{aligned}\beta_l^i &= GumbelSoftmax(\alpha_l^i|\alpha_l) \\ &= \exp[\alpha_l^i + n_l^i/\tau] \Big/ \sum_i \exp[\alpha_l^i + n_l^i/\tau] \end{aligned} \tag{6}$$

where, $\alpha_l = g_l(F_l)$ is the gate signal generated by the nonlinear function $g_l(\cdot)$ in F_l. $\beta_l^i \in \{0, 1\}$. $n_l^i \sim Gumbel(0, 1)$ is a random sampling of Gumbel distribution. τ is a temperature parameter that affects the gumbelsoftmax function.

IV. EXPERIMENT

A. SETUP

1) EXPERIMENTAL HARDWARE SPECIFICATION AND IMPLEMENTATION DETAILS

The experiment in this paper was implemented in MS COCO 2017 [70]. COCO contains 80 categories. COCO trainval35k split (118K image) was used for training, and minimal set (5K image) was used as the verification of this study. Standard average precision (AP), AP_{50}, AP_{75}, AP_S, AP_M and AP_L are used to evaluate the model performance. Our work is based on the Faster RCNN and ideas of Transformer, whose backbone mainly is ResNet, in order to compare with more general models and SOTAs(mainly the backbones are ResNet and ResNeXt [71]), we chose ResNet and ResNeXt as the backbone for fair comparison. The backbones mentioned above are pre-trained networks on ImageNet [72], and then the whole networks were finetuned and the backbones' parameters on the training set were frozen. For fair comparison, the size of the input images is resized to 800 pixels or 1000 pixels for shorter and longer edges, respectively.

For all experiments, we use SGD optimizer to train our models end-to-end for 12 epochs on a machine, whose CPU is Intel i7-9700k, 32 RAM, 4 NVIDIA GeForce GTX TITAN X GPUs with SBN [73] and the CUDA version is 10.1. The deep learning framework is Pytorch 1.7.1. Linear warm-up strategy for 500 iterations is leveraged at the beginning of training. Each mini-batch contains 2 images of each GPU and 512 regions of interest (ROI) of each image, and the positive and negative ratio is 1:3. We initialize the learning rate as 0.01 and decrease to 0.001 and 0.0001 at 8th-epoch and 11th-epoch. The momentum is set as 0.9 and the weight decay is 0.0001. An end-to-end region proposal network (RPN) [43] is used to generate region proposals. In order to make the model more robust, some data enhancement methods are used, such as geometric distortion, color jitter and so on.

2) HYPER-PARAMETERS

As for the hyper-parameters of the transformer module, $1/\sqrt{d_k}$ in **Equation 1** was set as 0.1. N in **Equation 5** was set as 4 and τ in **Equation 6** was a learned parameter, the Gumbel-Softmax distribution can adaptively adjust the "confidence" of proposed samples during the training process. We set it as 0.1 initially because it should approach to 0 and $\tau > 0$, at higher temperatures, Gumbel-Softmax samples are no longer one-hot, and become uniform as $\tau \to \infty$.

We apply the DropBlock [68] to each transformed feature map, to alleviate the over-fitting problem. Follow [59], we set block size = 5 and keep prob = 0.9.

3) LOSS FUNCTION

To reduce the computational complexity of the model and save resources, the loss function not only contains the classification and regression losses, but also adds the computation cost as a loss constraint [72] to achieve the optimal balance between the detection accuracy and the amount of calculation.

$$L_C = ((C_R - C_{target})/(C_{max} - C_{min}))^2 \tag{7}$$

$$C_{target} = C_{min} + \alpha \cdot (C_{max} - C_{min}) \tag{8}$$

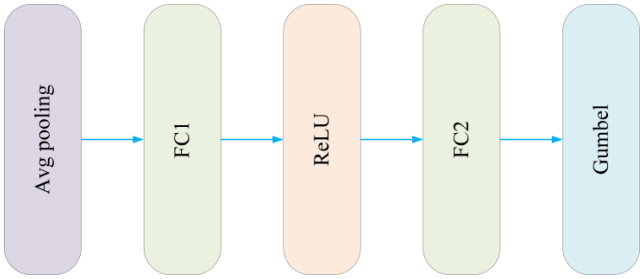


FIGURE 5. CNN gate block.

C_{max}, C_{min} represent the computation cost of the highest configuration and the lowest configuration respectively, and C_R represents the actual computation cost. C_{target} is controlled by super parameter α. The final loss function is as follows,

$$\begin{aligned} L(\{p_i\}, \{t_i\}) = {}& 1/N_{cls} \sum_i L_{cls}(p_i, p_i^*) \\ & + 1/N_{reg} \sum_i p_i^* L_{reg}(t_i, t_i^*) + \lambda L_C \end{aligned} \tag{9}$$

where i is the index of an anchor in a mini-batch, and p_i is the prediction probability that the anchor i is a target. If the anchor is positive, $p_i^* = 1$, otherwise $p_i^* = 0$. The 4-dimensional vector t_i representing the four angular coordinates of the prediction box and t_i^* is the coordinate vector of the truth bounding box. L_{cls} is the log loss on two categories (target and non-target). The regression loss $L_{reg}(t_i, t_i^*) = L_1(t_i - t_i^*)$ is a smooth L1 function. This term $p_i^* L_{reg}$ indicates that the regression function is only activated at $p_i^* = 1$. These two terms of L_{cls}, L_{reg} and L_C are balanced by the balance parameters λ.

B. COMPARISON

We compared TDFP with the most advanced object detectors in the test of MS-COCO [70] benchmark test-dev 2017. In these experiments, the images are randomly scaled from 640 pixels to 800 pixels in the training process, and the number of iterations is increased to 200K. We used the same settings and super parameters (e.g., learning rate, NMS

threshold, etc.) obtained from FPT [39] and DyFPN [67] for TDFP. Table 1 lists the comparison of the results of some detectors. R50, R101, RXt50, and RXt101 indicate ResNet50, ResNet101, ResNeXt50, and ResNeXt101.

Using resnext-101 as the backbone, the AP of TDFP reaches 46.2% and the AP_S is 26.3%. In the same backbone network, compared with Detr [28] based on Transformer, the detection accuracy of TDFP and large targets is slightly inferior, but the AP_S is 2.6%-2.9% higher than that of Detr and UP-DETR [29]. The more obvious result is that our method surpasses the transformed-based ViT-FRCNN [61] 6.0%-6.9%. We also found a surprising result, compared with the two-stage detector SpineNet [3], Faster RCNN [38], AugFPN [7], DyFPN [67] and the one-stage detector RetinaNet [47] based on CNN, the network TDFP has large improvement of AP, AP_{50}, AP_{75}, APM, AP_L on COCO.

C. ABLATION STUDY

The ablation study was performed on MS COCO 2017 val set, and the main backbone network was ResNet-50. The purpose of this study is as follows.

1) COMPARISION OF TRANSFORMER METHODS

In this section, we evaluate the importance of Transformer module (TS module), Single-scale transformation (SS TS) and cross-scale transformation (CS TS). As shown in Table 2, when the TS module is not added, the network fuses the features through the convolution layer, and the detection accuracy is the worst. The detection effect of transformer (TS) module is better than that of convolution. The AP of CS TS is 0.5% higher than that of SS TS, but the detection result of both CS TS and SS TS is the best. The AP of small target is 2.6% higher than that of no TS module.

Therefore, the transformed features have more abundant local and global context information to establish the relationship between features, as shown in Figure 6, which shows the visual comparison of features through convolution layer, single-scale transformation and cross-scale transformation. Among them, columns a, b, c, d and e are the original image, the convolution layer, the Single-scale transformation, the cross-scale transformation and the fusion feature maps after the Single-scale transformation and the cross-scale transformation. As can be seen from Figure 6, compared with the convolution layer, the self-attention layer can obtain more abundant global context feature information. cross-scale transformation can get more context information of multi-scale features than Single-scale transformation and realize the interaction between multi-scale features. Single-scale transformation and cross-scale transformation are used to capture the relationship between features with longer distance, and they are more sensitive to the features of small targets.

2) THE NECESSITY OF COMPUTATION LOSS

The influence of CC loss (Table 3) and resource limitation coefficient are studied. When CC loss is not used, the calculation amount is the largest. Although CC loss can lead to a small decrease in detection accuracy, it can greatly reduce the calculation amount and achieve a better balance between the accuracy and the calculation.

3) THE NECESSITY OF TRAINING STRATEGY

To explore the application effect of SBN [73] and DropBlock [68] in TDFP (as shown in Table 4), both SBN and DropBlock improve the model performance of TDFP, and their combination can achieve better results, making the bounding box AP improved by 1.6% - 2.1%.

TABLE 1. Comparison with SOTA (%).

Model	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CNN-based:							
SpineNet[3]	SpineNet-49	42.8	62.3	46.1	23.7	45.2	57.3
AugFPN[7]	R50-Faster RCNN	38.8	61,.5	42.0	23.3	42.1	47.7
AugFPN[7]	R101-Faster RCNN	40.6	63.2	44.0	24.0	44.1	51.0
Cascade CNN[18]	R101	42.8	62.1	46.3	23.7	45.5	55.2
FasterRCNN[38]	R50-FPN	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN[38]	R101-FPN	42.0	62.5	45.9	25.2	45.6	54.6
FPN[40]	R50-Faster RCNN	37.7	58.7	40.8	21.7	40.6	46.7
FPN[40]	R101-Faster RCNN	39.7	60.7	43.2	22.5	42.9	49.9
RetinaNet[47]	R101	41.1	60.6	44.4	23.5	44.3	52.3
DyFPN[67]	R50-FPN	39.0	60.7.	42.2	22.4	41.8	49.0
DyFPN[67]	R101	40.8	62.4	44.6	23.4	44.2	51.7
CNN and Transformer-based:							
DETR[28]	R50	42.0	62.4	44.2	20.5	45.8	61.1
DETR[28]	R101	43.5	63.8	46.4	21.9	48.0	61.8
BFP-FPN[39]	R101	37.9	59.6	40.1	19.5	41.0	53.5
BFP-FPT[39]	R101	41.6	60.9	44.0	23.4	42.5	53.1
Transformer-based:							
UP-DETR [29]	R50	42.8	63.0	45.3	20.8	47.1	61.7
BoTNet[33]	R50	42.1	-	-	22.5	-	59.1
BoTNet[33]	R101	43.3	-	-	24.2	-	60.7
ViT-FRCNN[61]	ViT-B/16-FRCNN	36.6	56.3	39.3	17.4	40.0	55.5
ViT-FRCNN[61]	ViT-B/16*-FRCNN	37.8	57.4	40.1	17.8	41.4	57.3
Ours:							
TDFP	R50	42.5	63.4	45.1	23.4	46.8	58.1
TDFP	R101	44.4	64.6	47.2	24.7	48.5	59.7
TDFP	RXt50	45.5	65.7	48.4	25.9	49.7	60.1
TDFP	RXt101	46.2	66.9	49.9	26.3	50.3	61.8

4) FPS AND GFLOPs

FLOPs measures model speed through theoretical calculations. FPS (frames per second) refers to the frequency of individual images that are displayed on a video device or the number of recorded images per second. We use torchscript models to measure FLOPs and FPS on an Nvidia GeForce RTX 2080 Ti GPU.

Under the same backbone ResNet-50, the comparison between TDFP and RetinaNet [47], Fast RCNN [38],

TABLE 2. Ablation studies of transformer module (TS module), single-scale transformation (SS TS) and cross-scale transformation (CS TS) (%).

TS	SS TS	CS TS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
×	-	-	40.4	60.5	40.8	20.8	43.3	53.2
√	√	×	39.7	59.5	41.7	21.4	43.3	53.8
√	×	√	40.5	60.9	42.3	21.9	44.1	54.7
√	√	√	42.5	63.4	45.1	23.4	46.8	58.1

TABLE 3. Ablation studies of calculated loss (CC loss).

CC loss	α	λ	GFLOPs(B)	AP(%)
×	–	–	345.2	42.1
√	0.5	0.1	248.0	41.9
√	0.3	0.1	236.7	41.8
√	0.2	0.1	187.9	40.6
√	0.2	0.5	172.1	40.5

TABLE 4. Ablation studies of SBN and DropBlock (%).

TDFP	SBN	DropBlock	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
√	×	×	40.4	60.5	40.8	20.8	43.3	53.2
√	√	×	40.9	61.2	42.1	21.8	44.8	54.1
√	×	√	40.6	61.0	41.5	21.2	43.9	53.9
√	√	√	42.5	63.4	45.1	23.4	46.8	58.1

TABLE 5. Comparison of FPS and GFLOPs of different detection methods.

Model	FPS/GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	18/205	38.7	58.0	41.5	23.3	42.3	50.3
EfficientDet[9]	6.5^{V100}/-	55.1	74.3	59/9	-	-	-
Faster RCNN	26/180	40.2	61.0	43.8	24.2	43.5	52.0
DETR	21/86	42.0	62.4	44.2	20.5	45.8	61.1
FCOS[49]	17/177	41.0	59.8	44.1	26.2	44.6	52.2
YOLOR[41]	30^{V100}/-	55.4	73.3	60.6	-	-	-
ScdYolov4[42]	16^{V100}/-	55.5	73.4	60.8	-	-	-
TDFP	24/188	42.6	62.4	44.1	24.3	45.8	55.1

Detr [28] as well as recent SOTAs in FPS and GFLOPs are reported in Table 5 (V100 represents NVIDIA TensorRT on a V100 GPU). When GFLOPs is close, TDFP model achieves the same result as Fast RCNN baseline. When the efficiency is not significantly reduced, FPS is higher than RetinaNet, which has lower AP_S, but greatly improves AP_L. Compared with Detr [28] which is based on transformer and CNN, FPS and GFLOPs are slightly inferior in terms of accuracy and overall accuracy, but the detection accuracy of small targets is greatly improved. And both FPS and GFLOPs are higher than FCOS [49].

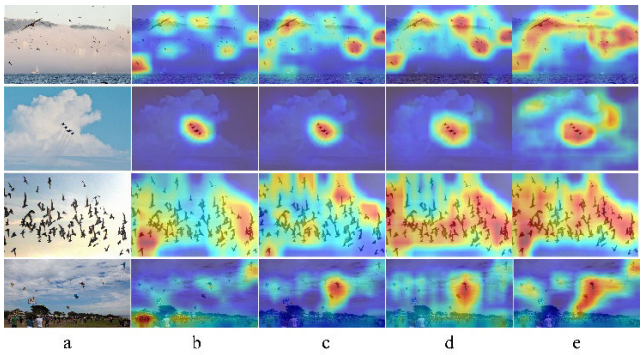


FIGURE 6. Features visualization.

FIGURE 7. Test results of COCO.

D. VISUALIZATION OF RESULTS AND DISCUSSION

In the test set of COCO, this paper selects some images which are difficult to detect, and the detection results are shown in Figure 7 On the whole, the detector in this paper can correctly detect the multi-scale targets in the image, and the detection results of small targets are also good.

Our model is to improve the detection accuracy of small targets combined CNN with Transformer methods. Both of them have advantages and disadvantages. The Transformer-based method has better detection results for large targets than small targets, while CNN is the opposite. Our method is to use the thought of Transformer in the process of constructing the feature pyramid. Compared with Faster RCNN, it may not be able to obtain better features of small targets. Compared with Detr, we proposed a better feature fusion and construct features. The rich feature pyramid combines local and global contextual information. Therefore, our method reached a compromise between Faster RCNN and Detr.

The experimental results show that the proposed model can effectively improve the accuracy of target detection and keep less computation. It achieves 44.4% and 46.2% AP on ResNet-101 and ResNeXt-101, respectively. The results surpass the previous two-stage detector Fast RCNN [38] and one-stage detector RetinaNet [47]. Compared with Detr [28], the overall detection accuracy in the same backbone network is lower, but it greatly improves the small target detection accuracy. At the same time, richer global information is beneficial to the big targets. Dynamic selection of the optimal multi-scale feature fusion method can obtain and aggregate more abundant multi-scale features and context information, which can better solve multi-scale problems, especially small targets. At the same time, the amount of calculation as a loss constraint training can reduce the amount of calculation without causing a significant decline in accuracy. In addition, the accuracy can be improved by a certain training strategy.

V. CONCLUSION AND FUTURE WORK

In order to mitigate the low accuracy problem of small targets due to less feature information in small targets and the limitations of CNN, a novel detection model based on CNN and transformer was proposed. In this model, a transformer module was designed to combine the local context and the global context information obtained by feature transformation. In this module, the method of dynamic multi-scale feature transformation and fusion determined by gate block was used to obtain optimal feature fusion and a feature pyramid with richer multi-scale feature information and context information. Through the above methods, the detection accuracy of small targets based on CNN and transformer is improved, and the detection results of large targets are better than that based on CNN. In addition, the proposed detection model takes the amount of calculation as a part of the loss function without significantly reducing the accuracy while reducing computation cost. However, this paper only applies the transformer idea to two-stage detector, which has a lot of optimization space in terms of accuracy and speed. In future work, we will consider combining transformer with one-stage detector or anchor-free detector.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable suggestions and corrections.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[2] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

[3] X. Du, T.-Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X. Song, "SpineNet: Learning scale-permuted backbone for recognition and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11592–11601.

[4] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 157–165.

[5] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019, *arXiv:1905.05055*. [Online]. Available: http://arxiv.org/abs/1905.05055

[6] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, and S.-J. Ko, "Parallel feature pyramid network for object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 234–250.

[7] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AugFPN: Improving multi-scale feature learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12595–12604.

[8] T. Kong, F. Sun, C. Tan, H. Liu, and W. Huang, "Deep feature pyramid reconfiguration for object detection," in *Proc. 15th Eur. Conf.*, Munich, Germany, Sep. 2018, pp. 169–185, doi: 10.1007/978-3-030-01228-1_11.

[9] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.

[10] Q. Zheng and Y. Chen, "Interactive multi-scale feature representation enhancement for small object detection," *Image Vis. Comput.*, vol. 108, Apr. 2021, Art. no. 104128.

[11] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2874–2883.

[12] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[13] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, no. 3, Jun. 2018, pp. 3588–3597.

[14] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global context networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 24, 2020, doi: 10.1109/TPAMI.2020.3047209.

[15] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection–SNIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3578–3587.

[16] L. Cui, R. Ma, P. Lv, X. Jiang, Z. Gao, B. Zhou, and M. Xu, "MDSSD: Multi-scale deconvolutional single shot detector for small objects," *Sci. China Inf. Sci.*, vol. 63, no. 2, pp. 98–100, Feb. 2020.

[17] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 1–9.

[18] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[19] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1222–1230.

[20] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6054–6063.

[21] K. Fukushima and S. Miyake, "Neocognitron: A selforganizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*. Springer, 1982.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762

[24] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.

[25] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *Proc. 37th Int. Conf. Mach. Learn., (ICML)*, vol. 119. Virtual Event, Jul. 2020, pp. 1691–1703.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*. [Online]. Available: http://arxiv.org/abs/2010.11929

[27] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," 2021, *arXiv:2101.11986*. [Online]. Available: http://arxiv.org/abs/2101.11986

[28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020, *arXiv:2005.12872*. [Online]. Available: http://arxiv.org/abs/2005.12872

[29] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," 2020, *arXiv:2011.09094*. [Online]. Available: http://arxiv.org/abs/2011.09094

[30] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*. [Online]. Available: http://arxiv.org/abs/2010.04159

[31] M. Zheng, P. Gao, X. Wang, H. Li, and H. Dong, "End-to-end object detection with adaptive clustering transformer," 2020, *arXiv:2011.09315*. [Online]. Available: http://arxiv.org/abs/2011.09315

[32] Z. Sun, S. Cao, Y. Yang, and K. Kitani, "Rethinking transformer-based set prediction for object detection," 2020, *arXiv:2011.10881*. [Online]. Available: http://arxiv.org/abs/2011.10881

[33] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 16519–16529.

[34] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "MaX-DeepLab: End-to-end panoptic segmentation with mask transformers," 2020, *arXiv:2012.00759*. [Online]. Available: http://arxiv.org/abs/2012.00759

[35] B. Dong, F. Zeng, T. Wang, X. Zhang, and Y. Wei, "SOLQ: Segmenting objects by learning queries," 2021, *arXiv:2106.02351*. [Online]. Available: http://arxiv.org/abs/2106.02351

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[37] X. Du, B. Zoph, W.-C. Hung, and T.-Y. Lin, "Simple training strategies and model scaling for object detection," 2021, *arXiv:2107.00057*. [Online]. Available: http://arxiv.org/abs/2107.00057

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[39] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, May 2020, Art. no. 103910.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[41] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13029–13038.

[42] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "You only learn one representation: Unified network for multiple tasks," 2021, *arXiv:2105.04206*. [Online]. Available: http://arxiv.org/abs/2105.04206

[43] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: https://arxiv.org/abs/2004.10934

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 21–37.

[45] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: http://arxiv.org/abs/1701.06659

[46] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*. [Online]. Available: http://arxiv.org/abs/1712.00960

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[48] X. Lu, Q. Li, B. Li, and J. Yan, "MimicDet: Bridging the gap between one-stage and two-stage object detection," 2020, *arXiv:2009.11528*. [Online]. Available: http://arxiv.org/abs/2009.11528

[49] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.

[50] Q. Zhou, C. Yu, C. Shen, Z. Wang, and H. Li, "Object detection made simpler by eliminating heuristic NMS," 2021, *arXiv:2101.11782*. [Online]. Available: http://arxiv.org/abs/2101.11782

[51] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyound anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.

[52] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9657–9666.

[53] Y. Chen, Z. Zhang, Y. Cao, L. Wang, S. Lin, and H. Hu, "Reppoints v2: Verification meets regression for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–11.

[54] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian, "Corner proposal network for anchor-free, two-stage object detection," 2020, *arXiv:2007.13816*. [Online]. Available: http://arxiv.org/abs/2007.13816

[55] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9759–9768.

[56] X. Zhou, V. Koltun, and P. Krähenbühl, "Probabilistic two-stage detection," 2021, *arXiv:2103.07461*. [Online]. Available: http://arxiv.org/abs/2103.07461

[57] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," 2019, *arXiv:1911.03584*. [Online]. Available: http://arxiv.org/abs/1911.03584

[58] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," 2021, *arXiv:2101.01169*. [Online]. Available: http://arxiv.org/abs/2101.01169

[59] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua, and Q. Sun, "Feature pyramid transformer," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 323–339.

[60] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[61] J. Beal, E. Kim, E. Tzeng, D. H. Park, A. Zhai, and D. Kislyuk, "Toward transformer-based object detection," 2020, *arXiv:2012.09958*. [Online]. Available: http://arxiv.org/abs/2012.09958

[62] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted Windows," 2021, *arXiv:2103.14030*. [Online]. Available: http://arxiv.org/abs/2103.14030

[63] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," 2021, *arXiv:2107.00641*. [Online]. Available: http://arxiv.org/abs/2107.00641

[64] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 7373–7382.

[65] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[66] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.

[67] M. Zhu, K. Han, C. Yu, and Y. Wang, "Dynamic feature pyramid networks for object detection," 2020, *arXiv:2012.00779*. [Online]. Available: http://arxiv.org/abs/2012.00779

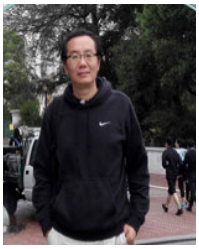
[68] G. Ghiasi, T. Y. Lin, and Q. V. Le, "DropBlock: A regularization method for convolutional networks," 2018, *arXiv:1810.12890*. [Online]. Available: https://arxiv.org/abs/1810.12890

[69] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel–Softmax," 2016, *arXiv:1611.01144*. [Online]. Available: http://arxiv.org/abs/1611.01144

[70] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "COCO-text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*. [Online]. Available: http://arxiv.org/abs/1601.07140

[71] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.

[72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[73] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7151–7160.



HONG LIANG (Member, IEEE) was born in Longchang, Sichuan, China. He is currently pursuing the Ph.D. degree. He is also a Professor. His research interests include computer vision and intelligent medical treatment.

YING YANG (Member, IEEE) was born in Xinxiang, Henan, China. She is currently pursuing the master's degree. Her research interest includes computer vision.

QIAN ZHANG (Member, IEEE) was born in Dongying, Shandong, China. She is currently pursuing the master's degree. She is also an Associate Professor. Her research interests include big data intelligent processing, computer vision, and intelligent medical treatment.

LINXIA FENG (Member, IEEE) was born in Jiujiang, Jiangxi, China. She is currently pursuing the master's degree. Her research interest includes computer vision.

JIE REN (Member, IEEE) was born in Jinan, Shandong, China. She is currently pursuing the master's degree. Her research interest includes computer vision.

QIYAO LIANG (Member, IEEE) was born in Yantai, Shandong, China. He is currently pursuing the master's degree. His research interest includes computer vision.