Centralized Feature Pyramid for Object Detection

Yu Quan, Dong Zhang®, Member, IEEE, Liyan Zhang®, and Jinhui Tang®, Senior Member, IEEE

Abstract—The visual feature pyramid has shown its superiority in both effectiveness and efficiency in a variety of applications. However, current methods overly focus on inter-layer feature interactions while disregarding the importance of intra-layer feature regulation. Despite some attempts to learn a compact intra-layer feature representation with the use of attention mechanisms or vision transformers, they overlook the crucial corner regions that are essential for dense prediction tasks. To address this problem, we propose a Centralized Feature Pyramid (CFP) network for object detection, which is based on a globally explicit centralized feature regulation. Specifically, we first propose a spatial explicit visual center scheme, where a lightweight feature is used to capture the globally long-range dependencies, and a parallel learnable visual center mechanism is used to capture the local corner regions of the input images. Based on this, we then propose a globally centralized regulation for the commonly-used feature pyramid in a top-down fashion, where the explicit visual center information obtained from the deepest intra-layer feature is used to regulate frontal shallow features. Compared to the existing feature pyramids, CFP not only has the ability to capture the global long-range dependencies but also efficiently obtain an all-round yet discriminative feature representation. Experimental results on the challenging MS-COCO validate that our proposed CFP can achieve consistent performance gains on the state-of-the-art YOLOv5 and YOLOX object detection baselines.

Index Terms—Feature pyramid, visual center, object detection, attention learning mechanism, long-range dependencies.

I. INTRODUCTION

OBJECT detection is one of the most fundamental yet challenging research tasks in the community of computer vision, which aims to predict a unique bounding box for each object of the input image that contains not only the location but also the category information [1]. In the past few years, this task has been extensively developed and applied to a wide range of potential applications, e.g., autonomous driving [2] and computer-aided diagnosis [3].

The successful object detection methods are mainly based on the Convolutional Neural Networks (CNNs) as the backbone followed by a two-stage (e.g., Fast/Faster R-CNN [4], [5]) or single-stage (e.g., SSD [6] and YOLO [7]) framework. However, due to the uncertainty of object sizes, a single feature scale cannot meet the requirements of high-accuracy recognition performance [8], [9]. To this end, methods (e.g., SSD [6] and FFP [10]) based on the in-network feature pyramid are proposed and achieve satisfactory results effectively and efficiently. The unified principle behind these methods is to assign a region of interest for each object of different size with the appropriate contextual information and enable these objects to be recognized in different feature layers.

The interaction between pixels or objects is crucial in image recognition tasks [11]. Effective feature interactions allow image features to obtain a wider and richer representation, enabling object detection models to learn the implicit relationship (such as favorable co-occurrence features [12], [13]) between pixels or objects. This has been shown to improve visual recognition performance [14], [15], [16], [17], [18], [19], [20]. For example, FPN [19] proposes a top-down inter-layer feature interaction mechanism, which enables shallow features to obtain global contextual information and semantic representations of deep features. NAS-FPN [15] tries to learn the network structure of the feature pyramid part via a network architecture search strategy and obtains a scalable feature representation. In addition to inter-layer interactions inspired by the non-local or self-attention mechanism [21], [22], finer intra-layer interactions for spatial feature regulation are also applied to object detection tasks, such as non-local features [23] and GCNet [24]. FPT [17] further integrates inter-layer cross-layer and intra-layer cross-space feature regulation methods and achieves exceptional performance.

Despite their success in object detection, the above methods are limited by the inherent small receptive fields of CNNs’ backbones [9]. As shown in Figure 1 (a), the standard CNNs’ backbone features can only locate those most discriminative object regions (e.g., the “body of an airplane” and the “motorcycle pedals”). To solve this problem, vision transformer-based object detection methods [25], [26], [27], [28] have been recently proposed and flourished. These methods first divide the input image into different image patches and then use the multi-head attention-based feature interaction among patches to complete the purpose of obtaining the global long-range dependencies. As expected, the feature pyramid is also employed in a vision transformer, e.g., PVT [28] and Swin Transformer [27]. Although these methods can address the limited receptive fields and the local contextual information in CNNs, an obvious drawback is their large computational complexity. For example,
a Swin-B [27] has almost 3× model FLOPs (i.e., 47.0 G vs. 16.0 G) than a performance-comparable CNNs model RegNetY [29] with the input size of 224 × 224. Besides, as shown in Figure 1 (b), since vision transformer-based methods are implemented in an omnidirectional and unbiased learning pattern, which is easy to ignore some corner regions (e.g., the “airplane engine,” the “motorcycle wheel” and the “bat”) that are important for dense prediction tasks. These drawbacks are more obvious in large-scale input images. Our research raises the question: is it necessary to use transformers on all layers? By analyzing shallow features, we argue that this may not be necessary. Research of the advanced methods [30], [31], [32] shows that the shallow features mainly contain some general object feature patterns, e.g., texture, color, and orientation, which are often not global. In contrast, the deep features reflect object-specific information, which usually requires global information [33], [34]. Therefore, we gracefully argue that a transformer may not be necessary for all layers.

In this work, we propose a Centralized Feature Pyramid (CFP) network for object detection, which is based on a globally explicit centralized regulation scheme. Specifically, based on a visual feature pyramid extracted from the CNNs backbone, we first propose an explicit visual center scheme, where an improved lightweight MLP architecture is used to capture the long-range dependencies and a parallel learnable visual center mechanism is used to aggregate the local key regions of the input images. Considering the fact that the deepest features usually contain the most abstract feature representations scarce in the shallow features [35], based on the proposed regulation scheme, we then propose a globally centralized regulation for the extracted feature pyramid in a top-down manner, where the spatial explicit visual center obtained from the deepest features are used to regulate all the frontal shallow features simultaneously. Compared to the existing feature pyramids, as shown in Figure 1 (c), CFP not only has the ability to capture the global long-range dependencies but also efficiently obtain an all-around yet discriminative feature representation. To demonstrate the superiority, we conduct extensive experiments on the challenging MS-COCO dataset [36]. Results validate that our proposed CFP can achieve consistent performance gains on the state-of-the-art YOLOv5 [37] and YOLOX [38] object detection baselines.

Our contributions can be summarized as follows:

- We introduce a spatial explicit visual center scheme, incorporating an improved lightweight MLP for capturing global dependencies and a learnable visual center for aggregating local key regions.
- We propose a globally centralized regulation for the commonly used feature pyramid, utilizing the deepest features to regulate the shallow features.
- Our centralized feature pyramid network demonstrates consistent improvement on state-of-the-art object detection baselines.

II. RELATED WORK

A. Feature Pyramid in Computer Vision

Feature pyramids are essential building blocks in modern object recognition systems, capable of effectively and efficiently detecting objects of different scales. The concept was first introduced in SSD [6], where a hierarchical representation of multi-scale features was used to capture multi-scale information and improve recognition accuracy. FPN [19] further developed the concept by building a top-down hierarchy of features from bottom-up in-network feature maps. PANet [18] added a bottom-up pathway to share feature information between layers, allowing high-level features to obtain sufficient details from low-level features. NAS-FPN [15] used a spatial search strategy to connect layers in the feature pyramid, resulting in extensible feature information. M2Det [39] constructed a multi-stage feature pyramid to extract multi-scale and cross-level features. In general, feature pyramids allow for dealing with the multi-scale changes in object recognition without incurring high computational overhead and provide multi-scale feature representations, including high-resolution features. Our proposed method improves upon current methods by introducing an intra-layer feature regulation approach to the feature pyramid, addressing their shortcomings in this aspect.

B. Object Detection

Object detection is a critical computer vision task that aims to identify objects of interest within an image and provide a complete scene description, including object category and location. With the rapid progress in Convolutional Neural Networks (CNNs) [40], many object detection models have achieved remarkable advances. The existing methods can be broadly classified into two categories: two-stage and single-stage detectors. Two-stage object detectors [4], [5], [41], [42], [43] typically use a Region Proposal Network (RPN) to generate a set of region proposals, and then extract features of these proposals and perform the classification and regression process using a separate learning module. However, the requirement to store and repeatedly extract features of each proposal leads to high computational costs and hinders the capture of
global feature representations. To address this, single-stage detectors [6], [7], [44], [45] directly perform prediction and region classification by generating bounding boxes. These single-stage methods have a global perspective in the design of feature extraction and use the backbone network to extract feature maps from the entire image to predict each bounding box. In this work, we also adopt single-stage object detectors (i.e., YOLOv5 [37] and YOLOX [38]) as our baseline models. Our goal is to enhance the representation of the feature pyramid used in these detectors.

C. Attention Learning and Long-Range Dependency

CNNs [40] focus more on the representative learning of local regions. However, this local representation does not satisfy the requirement for global context and long-range dependencies of modern recognition systems. To this end, the attention learning mechanism [22] is proposed that focuses on decision making. For example, the non-local operation [21] uses the non-local neural network to directly capture long-range dependencies, demonstrating the significance of non-local modeling for tasks of video classification, object detection, and segmentation. However, the local representation of the internal nature of CNNs are not resolved, i.e., CNNs features can only capture limited contextual information. To address this problem, Transformer [22], which mainly benefits from the multi-head attention mechanism has caused a great sensation recently and achieved great success in the field of computer vision, such as image recognition [25], [26], [27], [46], [47], [48]. For example, the representative ViT divides the image into a sequence with position encoding and then uses the cascaded transformer block to extract the parameterized vector as the visual representation. On this basis, many excellent models [46], [49], [50] have been proposed through further improvement and have achieved good performance in various tasks of computer vision. Nevertheless, the transformer-based image recognition models still have the disadvantages of being computationally intensive and complex.

While transformer-based image recognition models have shown remarkable performance, they are computationally intensive and complex. To mitigate these limitations, recent studies [51], [52], [53], [54] have shown that replacing attention-based modules in transformer models with MLP can still achieve good performance. This is because MLP and attention mechanisms are both global information processing modules. The introduction of MLP-Mixer [51] into computer vision reduces changes to the data layout and can better capture long-range dependencies and spatial relationships through the interaction between spatial and channel features. Although MLP-style models have limitations in capturing fine-grained feature representations, they have the advantage of a simpler network structure than transformers. In our work, we also use MLP to capture global contextual information and long-range dependencies in images. Our contribution lies in using a proposed spatial explicit visual center scheme to centralize the information captured. Besides, our other main contribution is the improvement of the existing MLP architecture, proposing a more lightweight MLP. The experimental results validate its effectiveness and efficiency.

III. OUR APPROACH

In this section, we introduce the proposed centralized feature pyramid (CFP) implementation details. We first make an overview architecture description for CFP in Section III-A. Then, we show the implementation details of the explicit visual center in Section III-B. Finally, we show how to implement the explicit visual center on an image feature pyramid and propose our global centralized regulation in Section III-C.

A. Centralized Feature Pyramid (CFP)

Although the existing methods have been largely concentrated on the inter-layer feature interactions, they ignore the intra-layer feature regulations, which have been empirically proven beneficial to vision recognition tasks. In our work, inspired by the previous works on dense prediction tasks [51], [53], [55], we propose a CFP for object detection, which is based on the globally explicit centralized intra-layer feature regulation. Compared to the existing feature pyramids, our proposed CFP can capture the global long-range dependencies and enable comprehensive and discriminative feature representations. As illustrated in Figure 2, CFP mainly consists of the following parts: the input image, a CNNs backbone used to extract the vision feature pyramid, the proposed Explicit Visual Center (EVC), the proposed Global Centralized Regulation (GCR) and a decoupled head network (which consists of a classification loss, a regression loss, and a segmentation loss) for object detection. In Figure 2, EVC and GCR is implemented on the extracted feature pyramid.

Concretely, we first feed an arbitrary RGB image into the backbone network (i.e., the Modified CSP v5 [56] and ResNet [57]) to extract a five-level one feature pyramid $X$, where the spatial size of each layer of features $X_i$ ($i = 0, 1, 2, 3, 4$) is $1/2, 1/4, 1/8, 1/16, 1/32$ of the input image, respectively. Based on the extracted feature pyramid, our CFP can be implemented on it. A lightweight MLP architecture is proposed to capture the global long-range feature dependencies based on $X_4$, where an MLP layer replaces the multi-head self-attention module of a standard transformer encoder. Compared to the transformer encoder based on the multi-head attention mechanism, our lightweight MLP architecture is simple in structure and has a lighter volume and higher computational efficiency (cf. Section III-B). Besides, a learnable visual center mechanism, along with the lightweight MLP, is used to aggregate the local corner regions of the input image. We name the above parallel structure network as the spatial EVC, which is implemented on the top layer (i.e., $X_4$) of the feature pyramid. Based on the proposed ECV, to enable the shallow layer features of the feature pyramid to benefit from the centralized visual information of the deepest feature at the same time, in an efficient pattern, we then propose a GCR in a top-down fashion, where the explicit visual center information obtained from the deepest intra-layer feature is used to regulate all the frontal shallow features (i.e., $X_3$ to $X_1$) simultaneously. Finally, we aggregate these features into a decoupled head network for instance classification and bounding-box regression.
the channel size is set to 256 in our work following \[19\]. At the same time, to reserve the local corner regions (i.e., feature maps are grouped along the channel dimension), we propose a learnable vision center mechanism implemented on \(X_i\) to aggregate the intra-layer local region features. The result feature maps of these two blocks are concatenated together along the channel dimension as the output of EVC for the downstream recognition. Next, instead of mapping the original features directly like \[37\], we added a Stem block between \(X_4\) and EVC for features smoothing. The Stem block consists of a \(7 \times 7\) convolution with an output channel size of 256, followed by a batch normalization layer and an activation function layer. The above processes can be formulated as:

\[
X = \text{Cat}(\text{MLP}(X_{\text{in}}); \text{LVC}(X_{\text{in}})),
\]

where \(X\) is the output of EVC, \(\text{Cat}(\cdot)\) denotes the feature map concatenation along the channel dimension, \(\text{MLP}(X_{\text{in}})\) and \(\text{LVC}(X_{\text{in}})\) denote the output features of the used lightweight MLP and the learnable visual center mechanism, respectively. \(X_{\text{in}}\) is the output of the Stem block, which is obtained by:

\[
X_{\text{in}} = \sigma(\text{BN}(\text{Conv}_{7, 7}(X_4))未来的.),
\]

where \(\text{Conv}_{7, 7}(\cdot)\) denotes a \(7 \times 7\) convolution with stride 1 and the channel size is set to 256 in our work following \[19\], \(\text{BN}(\cdot)\) denotes a batch normalization layer, and \(\sigma(\cdot)\) denotes the ReLU activation function.

1) Lightweight MLP: The proposed lightweight MLP mainly consists of two modules: a depthwise convolution-based module \[58\] and a channel MLP-based block, where the input of the MLP-based module is the output of the depthwise convolution-based \[51\] module. These two blocks are both followed by a channel normalization \(\text{GN}(\cdot)\) and a residual connection \(\text{DConv}(\cdot)\) to improve the feature generalization and robustness ability. Specifically, for the depthwise convolution-based module, features output from the Stem module \(X_{\text{in}}\) are first fed into a depthwise convolution layer, which has been processed by a group normalization \(\text{GN}(\cdot)\) and then the channel MLP \[51\] is implemented. The above processes can be formulated as:

\[
\tilde{X}_{\text{in}} = \text{DConv}(\text{GN}(X_{\text{in}})) + X_{\text{in}},
\]

where \(\tilde{X}_{\text{in}}\) is the output of the depthwise convolution-based module, \(\text{GN}(\cdot)\) is the group normalization, and \(\text{DConv}(\cdot)\) is a depthwise convolution \[58\] with a kernel size of \(1 \times 1\).

For the channel MLP-based module, features output from the depthwise convolution-based module \(\tilde{X}_{\text{in}}\) are first fed to the a group normalization, and then the channel MLP \[51\] is implemented on these features. Compared to space MLP, channel MLP can not only effectively reduce the computational complexity but also meet the requirements of general vision tasks \[38\], \[45\]. After that, channel scaling and DropPath are implemented. After that, a residual connection of \(\tilde{X}_{\text{in}}\) is implemented. The above processes can be formulated as:

\[
\tilde{X}_{\text{in}} = \text{DConv}(\text{GN}(X_{\text{in}})) + X_{\text{in}},
\]

where \(\tilde{X}_{\text{in}}\) is the output of the lightweight MLP-based module.

**B. Explicit Visual Center (EVC)**

As illustrated in Figure 3, our proposed EVC mainly consists of two blocks connected in parallel, where a lightweight MLP is used to capture the global long-range dependencies (i.e., the global information) of the top-level features \(X_4\). At the same time, to reserve the local corner regions (i.e., the local information), we propose a learnable vision center mechanism implemented on \(X_i\) to aggregate the intra-layer local region features. The result feature maps of these two blocks are concatenated together along the channel dimension as the output of EVC for the downstream recognition.
The above processes are expressed as:

\[
\text{MLP}(X_{in}) = \text{CMLP}(\text{GN}(\tilde{X}_{in})) + \tilde{X}_{in}, \quad (4)
\]

where \(\text{CMLP}(\cdot)\) is the channel MLP [51]. In our paper, for the presentation convenience, we omit channel scaling and drop-path in Eq. 3 and Eq. 4.

2) **Learnable Visual Center (LVC):** LVC maps the input feature \(X_{in}\) of shape \(C \times H \times W\) to a set of \(C\)-dimensional features \(\tilde{X}_{in} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N\}\), where \(N = H \times W\) is the total number of the input features. Next, \(\tilde{X}_{in}\) will learn an inherent codebook \(B = \{b_1, b_2, \ldots, b_K\}\) containing two important components: 1) \(K\) number of codewords (i.e. visual centers); 2) a set of smoothing factors \(S = \{s_1, s_2, \ldots, s_K\}\) for the visual centers. Specifically, features from the Stem block \(X_{in}\) are first encoded by a combination of a set of convolution layers (which consist of a \(1 \times 1\) convolution, a \(3 \times 3\) convolution, and a \(1 \times 1\) convolution). It is worth noting that the first \(1 \times 1\) is used to reduce the channel of the input features, and the last \(1 \times 1\) is used to increase the channel of the output features, while the middle \(3 \times 3\) layer has a smaller channel for the input and output features. Thus, this set of convolution layers not only reduces the computational volume but also allows the model to converge better. Next, the encoded features are processed by a CBR block, consisting of a \(3 \times 3\) convolution with a BN layer and a ReLU activation function. Through the above steps, the encoded features \(\tilde{X}_{in}\) are entered into the codebook. the learned smoothing factors \(s_i\) \((i = 1, 2, \ldots, K)\) can be used in codebook to match \(\tilde{X}_i\) \((i = 1, 2, \ldots, N)\) with the corresponding \(b_K\) \((i = 1, 2, \ldots, K)\). Then, the learnable weights are generated based on the difference between the two, and the weights are multiplied over the difference to sum up, and the final output is a \(C\)-dimensional vector \(e\) (as shown in Eq. 5 and Eq. 6). We can selectively focus on category-related feature map information. The information of the whole image with respect to the \(k\)-th codeword can be calculated by:

\[
\mathbf{e}_k = \sum_{i=1}^{N} s_i \frac{e^{-s_i \| \tilde{x}_i - b_k \|_2}}{\sum_{j=1}^{K} e^{-s_j \| \tilde{x}_i - b_j \|_2}} \left(\tilde{x}_i - b_k\right), \quad (5)
\]

where \(\tilde{x}_i - b_k\) is the difference between the \(N\) \(C\)-dimensional feature vectors and the \(K\) codeword vectors. Then, the learnable weight \(\frac{e^{-s_i \| \tilde{x}_i - b_k \|_2}}{\sum_{j=1}^{K} e^{-s_j \| \tilde{x}_i - b_j \|_2}}\) is generated based on the \(\tilde{x}_i - b_k\), where the \(\| \cdot \|_2\) denotes L2 parametric operation, \(s_i\) denotes the smoothing factor of the \(k\)-th codeword, and \(\| \tilde{x}_i - b_k \|_2^2\) is the output value of the \(k\)-th codeword. Next, the weights are multiplied by the \(\tilde{x}_i - b_k\) to obtain information about the
position of a pixel relative to a codeword. Finally, We sum these N results to obtain \( \mathbf{e}_k \), which is information about the whole image relative to the \( k \)-th codeword. After that, we use \( \phi \) to fuse all \( \mathbf{e}_k \), where \( \phi \) contains BN layer with ReLU and the mean layer. Finally, we sum over the \( K \) results to obtain \( \mathbf{e} \), which is the full information of the whole image concerning the K codewords. Both \( \mathbf{e}_k \) and \( \mathbf{e} \) are \( C \)-dimensional vectors.

\[
\mathbf{e} = \sum_{k=1}^{K} \phi(\mathbf{e}_k).
\]

A set of impact factors is then predicted to highlight the categories that need to be highlighted. First, we used FC to map the feature \( \mathbf{e} \) to \( C \times 1 \times 1 \) size as the impact factor. After that, we use the channel-wise multiplication between the input features \( \mathbf{X}_{in} \) from the Stem block and the impact factor coefficient \( \delta(\cdot) \). The feature representations, after being highlighted, are obtained as follows:

\[
\mathbf{Z} = \mathbf{X}_{in} \otimes (\delta(\mathbf{FC}(\mathbf{e}))),
\]

where FC denotes a fully connected layer, and \( \delta(\cdot) \) is the sigmoid function. \( \otimes \) is channel-wise multiplication. Finally, we perform a channel-wise addition between features \( \mathbf{X}_{in} \) output from the Stem block and the local corner region features \( \mathbf{Z} \), which is formulated as:

\[
\text{LVC}(\mathbf{X}_{in}) = \mathbf{X}_{in} \oplus \mathbf{Z},
\]

where \( \oplus \) is the channel-wise addition.

C. Global Centralized Regulation (GCR)

EVC is a generalized intra-layer feature regulation method that can extract global long-range dependencies and preserve the local corner regional information of the input image as much as possible, which is very important for dense prediction tasks. However, using EVC at every level of the feature pyramid would result in a large computational overhead. To improve the computational efficiency of intra-layer feature regulation, we further propose a GCR for a feature pyramid in a top-down manner. Specifically, as illustrated in Figure 2, because the deepest features usually contain the most abstract feature representations scarce in the shallow features [35], [60], our spatial EVC is first implemented on the top layer (i.e., \( \mathbf{X}_4 \)) of the feature pyramid. Then, the obtained features \( \mathbf{X} \), which includes the spatial explicit visual centers, are used to regulate all the frontal shallow features (i.e., \( \mathbf{X}_3 \) to \( \mathbf{X}_1 \)) simultaneously. In our implementation, on each corresponding low-level features, the features obtained in the deep layer are upsampled to the same spatial scale as the low-level features and then are concatenated along the channel dimension. Based on this, the concatenated features are downsampled by a \( 1 \times 1 \) convolution into the channel size of 256 as [19]. In this way, we can explicitly increase the spatial weight of the global representations at each layer of the feature pyramid in the top-down path, such that our CFP can effectively achieve an all-around yet discriminative feature representation.

IV. EXPERIMENTS

A. Dataset and Evaluation Metrics

1) Dataset: In this work, Microsoft Common Objects in Context (MS-COCO) [36] is used to validate the superiority of our proposed CFP. MS-COCO contains 80 classes of the common scene objects, where the training set, val set, and test set contain 118k, 5k, and 20k images, respectively. In our experiments, for a fair comparison, all the training images are resized into a fixed size of \( 640 \times 640 \) as in [19]. For data augmentation, we adopt the commonly used Mosaic [45] and MixUp [61] in our experiments. Mosaic cannot only enrich the image data but also indirectly increase our batch size. MixUp can play a role in increasing the model generalization ability. In particular, following [38], our model turns the data augmentation strategy off at the last 15 epochs in training.

2) Evaluation Metrics: We mainly follow the commonly used object detection evaluation metric – Average Precision (AP) in our experiments, which include AP\(_0\), AP\(_5\), AP\(_s\), AP\(_p\), and AP\(_L\). Besides, to quantitative the model efficiency, GFLOPs, Frame Per Second (FPS), Latency, and parameters (Params.) are also used. In particular, following [38], Latency and FPS are measured without post-processing for a fair comparison.

B. Implementation Details

1) Baselines: To validate the generality of CFP, we use two state-of-the-art baseline models in our experiments: YOLOv5 [37] and YOLOX [38]. In our experiments, we use the end-to-end training strategy and employ their default training and inference settings unless otherwise stated.

- **YOLOv5** [37]. The backbone is a modified cross-stage partial network v5 [56] and DarkNet53 [44], where the modified cross-stage partial network v5 is used in the ablation study and the DarkNet53 is used in a result comparisons with the state-of-the-art. The neck network is FPN [19]. The object detection head is the coupled head network, which contains a classification branch and a regression branch. In YOLOv5, according to the scaling of network depth and width, three different scale networks are generated, they are YOLOv5-Small (YOLOv5-S), YOLOv5-Medium (YOLOv5-M), and YOLOv5-Large (YOLOv5-L).

- **YOLOX** [38]. Compared to YOLOv5, the whole network structure of YOLOX remains unchanged except for the coupled head network. In YOLOX, the object detection head is the decoupled head network.

2) Backbones: In our experiments, two backbones are used.

- **DarkNet53** [44]. DarkNet53 mainly consists of 53 convolutional layers (basically \( 1 \times 1 \) with \( 3 \times 3 \) convolutions), which is mainly used for the performance comparisons with state-of-the-art methods in Table IX.

- **Modified CSPNet v5** [37]. For a fair comparison, we choose YOLOv5 (i.e., the Modified CSPNet v5) as our backbone network. The output feature maps are the ones from stage5, which consists of three convolutions (Conv, BN and SiLU [62]) operations and a spatial pyramid pooling [63] layer (\( 5 \times 5 \), \( 9 \times 9 \) and \( 13 \times 13 \)).
3) Comparison Methods: We consider using MLP instead of attention-based, which performs well and is computationally less expensive. Therefore, we design a series of MLPs and attention-based variants. Through the ablation study, we choose an optimal variant for our LVC mechanism, as well as CFP approach, called lightweight MLP. Figure 4 (a) shows the PoolFormer structure [53], which consists of a Pooling operation sub-block and a two-layered MLP sub-block. Considering that the Pooling operation corrupts the detailed features, we choose some convolutions that are structurally lightweight and guarantee accuracy simultaneously. Therefore, we designate CPSLayer [56] and depthwise convolution as token mixers. They are called CSPM [53], [56] and MLP (Ours) in (c) and (e) of Figure 4, respectively. Compared with MLP variants, the structures (b), (d), and (f) are corresponding attention-based variants, respectively. It is worth noting that we choose the channel MLP in the MLP variants. Then, we use convolutional position-encoding to prevent the translational invariance caused by absolute position-encoding.

4) Training Settings: We first train our CFP on MS-COCO using pre-trained weights from the YOLOX [38] or YOLOv5 [37] backbone, where all other training parameters are similar in all models. Considering the local hardware condition, our model is trained for 150 epochs, including five epochs for learning rate warmup, as in [57]. We use 2 GeForce RTX 3090 GPUs with the Batch Size is 16. Our training settings remain largely consistent from the baseline to the final model. The input image training size is 640 × 640. The learning rate is set to lr × BatchSize / 64 (i.e., the linear scaling strategy [64]), where the initial learning rate is set to lr = 0.01 and the cosine lr schedule is used. The weight decay is set to 0.0005. The optimizer for the model training process selects stochastic gradient descent, where the momentum is set to 0.9. Besides, following [19], we evaluate the AP every ten training epochs and report the best one on the MS-COCO val set.

5) Inference Settings: For the inference of our model, the original image is scaled to the object size (640 × 640) and the rest of the image is filled with gray. Then, we feed the image into the trained model for detection. In the inference that FPS and Latency are all measured with FP16-precision and batch = 1 on a single GeForce RTX 3090. However, keep in mind that the inference speed of the models is often uncontrolled, as speed varies with software and hardware.

C. Ablation Study

Our ablation study aims to investigate the effectiveness of LVC, MLP, EVC, and CFP in object detection. To this end, we perform a series of experiments on the MS-COCO val set. From Table I, it can be seen that we analyze the effects of LVC, MLP, and EVC on the average precision, the number of parameters, computation volume, and Latency using YOLOv5-L [37] and YOLOX-L [38] as the baselines, respectively. We respectively present the feature visualization results of lightweight MLP, LVC, and EVC generated on the YOLOX-L [38] model, and carry out a detailed analysis of Figure 5. A detailed analysis of our MLP variants and attention-based variants in precision and Latency are presented in Table III, using YOLOX-L as the baseline. Table V shows the effect of the number of visual centers K on the LVC at the YOLOX-L baseline. From Table VI, we can intuitively see the impact of our CFP method on the model with the number of repetitions R at the YOLOX-L baseline.

1) Effectiveness on Different Baselines: In Table I, we perform ablation studies on the MS-COCO val set using YOLOv5-L [37] and YOLOX-L [38] as baselines for the
proposed MLP, LVC, and EVC, respectively. As shown in Table I, when we only use the LVC mechanism to aggregate local corner region features, and the parameters, computation volume, and Latency are all within the acceptable growth range, the mAP of our YOLOv5-L and YOLOX-L models are improved by 1.0% and 1.3%, respectively. Furthermore, when we capture the global long-range dependencies using only the lightweight MLP structure, the mAP of the YOLOv5-L [37] and YOLOX-L [38] models improve by 0.6% and 1.3%, respectively. Most importantly, when we use both LVC and MLP (the EVC scheme) on the YOLOv5-L and YOLOX-L baselines, the mAP of both models are improved by 1.4%. Further analysis shows that when EVC scheme is applied to the YOLOv5-L baseline and YOLOX-L baseline, respectively, mAP of the YOLOX-L model can be improved to 49.2%. Its parameter number and computation volume are lower than those of the YOLOv5-L model. The results show that the EVC scheme is more effective in the YOLOX-L baseline, and the overhead is slightly smaller than that of the YOLOv5-L baseline. YOLOX-L is used as the baseline in subsequent ablation experiments.

In Figure 5, we show two examples of feature visualization. Firstly, lightweight MLP can better capture long-range dependencies. For example, the edge information of “person” and “zebras” can be comprehensively captured. After “person” and “zebras” pass through the LVC module, the capture of local corner regions is very effective. In addition, it can be seen intuitively from the last column that “person” and “zebras” can realize the fusion of edge information and corner feature information through the EVC module.

2) Superiority of EVC: An intra-layer feature regulation from the perspective of inter-layer feature interactions and intra-layer feature regulations of feature pyramids, which makes up for the shortcomings of current methods in this regard. We conducted ablation experiments with inter-layer feature interactions represented by the feature pyramid approaches and intra-layer feature representations expressed by the attention-based methods, respectively. The experimental results are given in Table II. To ensure the fairness and consistency of the experiments. We uniformly used ResNet-50 as the backbone and MS-COCO val set as the dataset. In addition, we set the epoch to 50. The mAP, AP$_{50}$, AP$_{75}$, and Params are used as evaluation metrics in the experimental evaluation.

As shown in Table II, our CFP$_{YOLOX-L}$ performs significantly well compared to Non-Local [21] and DESTR [65] in the intra-layer feature representations ablation experiments. In comparison with MiViTv2 [66] and MaxViT [67], although CFP$_{YOLOX-L}$ has no advantage in accuracy, our model consume fewer memory resources. In inter-layer feature interactions, our CFP$_{YOLOX-L}$ performs significantly better. While reducing memory resource consumption, the mAP improves by 2.5% to 12.4%. The above results indicate that CFP provides significant improvement in object detection tasks through intra-layer feature representations and inter-layer feature interactions.

3) Comparisons With MLP Variants: Table III shows the detection performance of MLP and attention-based variants based on the YOLOX-L baseline on the MS-COCO val set. We first analyze the comparison results of MLP variants. We can observe that the PoolFormer structure obtains the same mAP (i.e., 47.80%) as the YOLOX-L [38] model. The performance of CSPM [53], [56] is even worse compared to the YOLOX-L model, which not only reduces the average precision by 0.1% but also increases the Latency by 0.74ms. But our proposed lightweight MLP structure improves the highest mAP (i.e., 49.10%) in the MLP variants, which is 1.3% better than the mAP of YOLOX-L. The result demonstrates that our choice of depthwise convolution as the token mixer in the MLP variant performs better. In the attention-based variants, the performance of PoolA, CSPA [22], [56], and DWA [22], [58] are all improved compared to YOLOX-L, and the mAP of DWA can reach 49.20%. But in fact, we compare the two best performing structures (MLP and DWA) find that the Latency of DWA increases by 2.84ms in the same hardware environment than MLP (Ours). From Table III, it can be found that our lightweight MLP is not only better but also faster in capturing long-range dependencies.

Besides, we also use YOLOX-L as the baseline, and an MLP-Mixer [51] module or an attention [22] module is built behind the backbone of YOLOX-L [38], respectively. Results are shown in Table IV. Comparison with the standard MLP-
Mixer [51] and attention-based methods shows that the mAP of the lightweight MLP increases by 0.6% to 1.0% with less computational costs. We can get the conclusion that our proposed lightweight MLP for capturing long-range dependencies of intra-layer features performed significantly.

4) Effect of K: As shown in Table V, we analyze the effect of the number of visual centers K on the performance of LVC. We choose YOLOX-L as the baseline, and with increasing K, we can observe that its performance shows an increasing trend. At the same time, the parameter number, computation volume, and the Latency of the model also tend to increase gradually. Notably, when K = 64, the mAP of the model can reach 49.10%, and when K = 128, the mAP of the model can get 49.20%. Although the performance of the model improves by 0.1% as K increases, its extra computational cost increases by 10.01G, and the corresponding inference time increases by 3.21ms. This may be because too many visual centers bring more redundant semantic information. Not only is the performance not significantly improved, but the computational effort is increased. So we choose K = 64.

5) Effect of R: From Table VI, we analyze the effect of the number of repetitions R of CFP on the performance. We still choose the YOLOX-L baseline, and as R increases, we can observe a trend of increasing and decreasing and then stabilizing the performance compared to the YOLOX-L model. Meanwhile, the number of parameters, computation volume, and Latency gradually increase. In particular, when R = 1, CFPYOLOX-L achieves the best performance mAP of 49.40%. When R = 2, the performance is instead reduced by 0.2% compared to R = 1. The reason may be that redundant feature information is not helpful for the task but increases the computational cost. Therefore, based on the above observations, we choose R = 1.

D. Efficiency Analysis

We analyze the performance of the MLP variants and attention-based variants from a multi-metric perspective. In Figure 6, all models take YOLOX-L [38] as a baseline and are trained on the MS-COCO [36] emvval set with the same data augmentation settings. Meanwhile, to demonstrate the effectiveness of the MLP structure, as shown in Table VII, we compare it with the state-of-the-art transformer methods and the MLP methods at this stage. As can be observed from Figure 6, we can intuitively see that the MLP (Ours) structure is significantly better than the other structures in terms of mAP, and it is lower than the other structures in terms of the number of parameters, computation volume, and the inference time. It can be shown that the MLP structure can guarantee
TABLE VII
RESULT COMPARISONS OF OUR LIGHTWEIGHT MLP WITH TRANSFORMER VARIANTS AND MLP VARIANTS METHODS

| Methods     | Backbone          | mAP (%) | Params. (M) | GFLOPs (G) |
|-------------|-------------------|---------|-------------|------------|
| MetaFormer [53] | PoolFormer-S12 [53] | 37.30   | 31.60       | 62.75      |
| MetaFormer [53] | PoolFormer-S24 [53] | 40.10   | 41.00       | 66.38      |
| MetaFormer [53] | PoolFormer-S36 [53] | 41.00   | 50.50       | 67.89      |
| Mask R-CNN [71] | AS-MLP-T [71]       | 46.00   | 48.00       | 117.45     |
| Mask R-CNN [71] | AS-MLP-S [71]       | 47.80   | 69.00       | 176.20     |
| DETR [25]     | ResNet-50 [19]      | 42.00   | 41.00       | 86.00      |
| YOLOx         | DeiT-base [46]      | 42.00   | 127.00      | 567.00     |
| ViDT (w/o Neck) [74] | Swin-base [27]   | 43.20   | 91.00       | 203.40     |
| REGO-Deformable DETR [72] | ResNet-50 [19]   | 43.80   | 40.00       | 173.00     |
| MLPYOLOX-L (Ours) | Modified CSP v5 [37] | 45.80   | 48.90       | 120.60     |
| MLPYOLOX-L (Ours) | Modified CSP v5 [37] | 49.10   | 56.57       | 163.22     |

TABLE VIII
RESULT COMPARISONS WITH YOLOv5 AND YOLOX. "†" IS OUR RE-IMPLEMENTATION RESULT. "-" DENOTES THAT THERE IS NO SUCH A SETTING

| Methods     | Backbone          | mAP (%) | AP50 (%) | AP75 (%) | APs (%) | APm (%) | APl (%) |
|-------------|-------------------|---------|----------|----------|---------|---------|---------|
| YOLOv5 [37] | Small [37]         | 35.50   | 55.30    | 47.10    | -       | -       | -       |
| YOLOv5 [37]† | Media [37]        | 42.70   | 62.30    | 48.00    | -       | -       | -       |
| YOLOx [37]  | Large [37]         | 45.20   | 64.10    | 49.20    | -       | -       | -       |
| CFPYOLOX [37] | Small [37]       | 36.00±0.5 | 56.20    | 47.80    | 22.80   | 42.90   | 51.60   |
| CFPYOLOX [37] | Media [37]       | 43.20±0.5 | 62.90    | 48.50    | 29.10   | 49.40   | 53.30   |
| CFPYOLOX [37] | Large [37]       | 46.60±1.4 | 64.90    | 50.00    | 30.40   | 51.70   | 59.50   |
| YOLOX [38]  | Small [38]         | 34.10   | 52.00    | 36.90    | 18.80   | 38.10   | 44.40   |
| YOLOX [38]† | Media [38]        | 45.60   | 64.30    | 48.90    | 28.00   | 50.20   | 59.70   |
| YOLOX [38]  | Large [38]         | 47.80   | 65.80    | 51.60    | 29.90   | 52.80   | 62.40   |
| CFPYOLOX [38] | Small [38]       | 41.10±7.0 | 60.00    | 44.30    | 24.20   | 45.40   | 54.50   |
| CFPYOLOX [38] | Media [38]       | 46.40±6.8 | 65.10    | 50.30    | 29.40   | 51.20   | 60.50   |
| CFPYOLOX [38] | Large [38]       | 49.40±1.6 | 67.90    | 53.40    | 31.50   | 54.80   | 64.20   |

Fig. 6. Multi-metrics comparison results between MLP variants and attention-based variants based on the MS-COCO [36] val set.

In Table VII, we give the comparative results of the MLP and transformer methods that are outstanding performers in object detection tasks at this stage. In the first half of Table VII, our MLPYOLOX-L method occupies less memory and has an average precision of 1.3% higher compared to Mask R-CNN (backbone as AS-MLP-S [71]). In the middle part of Table VII, we find that our MLPYOLOX-L can improve the mAP by up to 7.1% compared to the transformer method (DETR [25]) without extra computational cost. With the same mAP, the number of parameters of MLP is reduced by 62.43M compared to REGO-Deformable DETR [72]. Therefore, we can find that MLP not only has high precision but also takes up less memory compared to the transformer methods. All in all, our MLP has outstanding performance in capturing long-range feature dependencies.

E. Comparisons With State-of-the-Art Methods

As shown in Table VIII, we validate the proposed CFP method on the MS-COCO val set with YOLOv5 (Small, Media, and Large) and YOLOX (Small, Media, and Large) as baselines. In addition, the data in Table IX show the comparison results of our CFP method compared to the advanced single-stage and two-stage detectors. Finally, we offer some visual comparison plots in Figure 7.
1) Comparisons With YOLOv5 and YOLOX Baseline: As shown in Table VIII, when YOLOv5 is chosen as the baseline, the mAP of our CFP method is enhanced by 0.5%, 0.5%, and 1.4% on the Small, Media, and Large size models, respectively. When YOLOX [38] is used as the baseline, the mAP improves by 7.0%, 0.8%, and 1.6% on the backbone networks of different sizes. It is worth noting that the main reason why we choose YOLOv5 (an anchor mechanism) and YOLOX (an anchor-free mechanism), as the baseline, is that the reciprocity of these two models in terms of network structure can fully demonstrate the effectiveness of our CFP approach. Most importantly, compared with YOLOX, although the network structure of YOLOv5 has no advantage; the mAP of our CFP method can still obtain 46.60%. Meanwhile, our mAP reaches 49.40% at the YOLOX baseline. Moreover, the CFP_YOLOX on the small backbone network is improved by 7.0% over YOLOX [38]. The main reason for this is that the LVC in our CFP can enhance the feature representations of the local corner regions through visual centers at the pixel level.

2) Comparisons on Speed and Accuracy: We perform a series of comparisons on the MS-COCO val set with single-stage, and two-stage detectors, and the results are shown in Table IX. We can first see the two-stage object detection models, including the Faster R-CNN series with different backbone
networks, Mask R-CNN, and D2Det. Our CFP\textsubscript{YOLOX-L} model has significant advantages in precision, inference speed, and time. Immediately after, we divide the single-stage detection methods into three parts in chronological order and then analyze them. There is no doubt that the proposed CFP\textsubscript{YOLOX-L} method improves the mAP by up to 27.80% compared to YOLOv3-ultralytics\textsuperscript{[44]} and its previous detectors. With nearly the same average precision, the CFP\textsubscript{YOLOv5-M} inferred 1.5 times faster compared to the EfficientDet-D2 detector. And comparing CFP\textsubscript{YOLOX-L} with EfficientDet-D3, the average accuracy is improved by 1.9%, and the inference speed is 1.8 times higher. In addition, in comparison with YOLOv4\textsuperscript{[45]} series, it can be found that the mAP of CFP\textsubscript{YOLOv5-L} is improved by 2.7% compared to YOLOv4-CSP\textsuperscript{[80]}. Besides, we can see all scaled YOLOv5\textsuperscript{[37]} models, including YOLOv5-S\textsuperscript{[37]}, YOLOv5-M\textsuperscript{[37]}, and YOLOv5-L\textsuperscript{[37]}. The average precision of its best YOLOv5-L\textsuperscript{[37]} model is 1.4% lower than the CFP\textsubscript{YOLOv5-L}. In the same way, our CFP method obtains a maximum average accuracy of 49.40%, which is 1.6% higher than YOLOX-L.

3) Qualitative Results on MS-COCO\textsuperscript{[36]} 2017 test Set: In addition, we also show in Figure 7 some visualization results of baseline (YOLOX-L\textsuperscript{[38]}), EVC\textsubscript{YOLOX-L}, and CFP\textsubscript{YOLOX-L} on the MS-COCOCO\textsuperscript{[36]} test set. It is worth noting that we use white, red, and orange boxes to mark where the detection task failures, respectively. White boxes indicate misses due to occlusion, light influence, or small object size. Red boxes indicate detection errors due to insufficient contextual semantic relationships, e.g., causing one object to be detected as two objects. The yellow boxes indicate an error in the object classification. In the first line, the detection result of YOLOX-L in part marked in the white box is not ideal due to the distance factor of “zebra”. And the EVC\textsubscript{YOLOX-L} can partially detect the “zebra” at a distance. Therefore, it is intuitively proved that EVC is very effective for small object detection in some intensive detection tasks. In the second line of the figure, YOLOX-L does not fully detect the “Cups” in the cabinet due to factors such as occlusion and illumination. The EVC\textsubscript{YOLOX-L} model alleviates this problem by using MLP structures to capture the long-range dependencies of the features in the object. Finally, the CFP\textsubscript{YOLOX-L} model uses the GCR-assisted EVC scheme and gets better results. In the third line of the figure, the CFP\textsubscript{YOLOX-L} model performs better in complex scenarios. Based on the EVC scheme, GCR is used to adjust intra-layer features for top-down, and CFP\textsubscript{YOLOX-L} can solve the problem of classification better.

V. CONCLUSION AND FUTURE WORK

In this work, we proposed a CFP for object detection, which was based on a globally explicit centralized feature regulation. We first proposed a spatial explicit visual center scheme, where a lightweight MLP was used to capture the globally long-range dependencies and a parallel learnable visual center was used to capture the local corner regions of the input images. Based on the proposed EVC, we then proposed a GCR for a feature pyramid in a top-down manner, where the explicit visual center information obtained from the deepest intra-layer feature was used to regulate all frontal shallow features. Compared to the existing methods, CFP not only has the ability to capture the global long-range dependencies, but also efficiently obtain an all-round yet discriminative feature representation. Experimental results on MS-COCO dataset verified that our CFP can achieve the consistent performance gains on the state-of-the-art object detection baselines. CFP is a generalized approach that can not only extract global long-range dependencies of the intra-layer features but also preserve the local corner
regional information as much as possible, which is very important for dense prediction tasks. Therefore, in the future, we will start to develop some advanced intra-layer feature regulate methods to improve the feature representation ability further. Besides, we will try to apply EVC and GCR to other feature pyramid-based computer vision tasks, e.g., semantic segmentation, object localization, instance segmentation and person re-identification.

ACKNOWLEDGMENT

The code and weights have been released at: CFPNet.

REFERENCES

[1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[2] M. Tremi et al., “Speeding up semantic segmentation for autonomous driving,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2020, pp. 1–17.

[3] M. Havaei et al., “Brain tumor segmentation with deep neural networks,” Med. Image Anal., vol. 35, pp. 18–31, Jan. 2017.

[4] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[5] 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.

[6] W. Liu et al., “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 1–17.

[7] 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.

[8] D. Zhang, L. Zhang, and J. Tang, “Augmented FCN: Rethinking context modeling for semantic segmentation,” Sci. China Inf. Sci., vol. 66, no. 4, pp. 1202–1215, Apr. 2015.

[9] D. Zhang, J. Tang, and K.-T. Cheng, “Graph reasoning transformer for image parsing,” in Proc. 30th ACM Int. Conf. Multimedia, Oct. 2022, pp. 2380–2389.

[10] P. Dollár, R. Appel, S. Belongie, and P. Perona, “Fast feature pyramids for object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[11] A. Vaswani et al., “Attention is all you need,” in Proc. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 1–11.

[12] Z. Luo, A. Mishra, A. Achkar, J. Eichel, S. Li, and P.-M. Jodoin, “Non-local deep features for salient object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 6593–6601.

[13] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, “GCNet: Non-local networks meet squeeze-excitation networks and beyond,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 1971–1980.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 213–229.

[15] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, “Designing network design spaces,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10425–10433.

[16] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2921–2929.

[17] L. Tu, Y. Zhan, B. Yu, and B. Du, “Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers,” 2022, arXiv:2203.02696.

[18] R. Li, Z. Mai, C. Trabelsi, Z. Zhang, J. Jang, and S. Sanner, “TransCAM: Transformer attention-based CAM refinement for weakly supervised semantic segmentation,” 2022, arXiv:2203.07239.

[19] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, “Segmentor: Transformer for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 7324–7325.

[20] F. Zhu, Y. Zhu, L. Zhang, C. Wu, Y. Fu, and M. Li, “A unified efficient pyramid transformer for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW), 2021, pp. 2667–2677.

[21] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, “Self-regulation for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 6933–6943.

[22] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 213–229.

[23] G. Jocher et al., “UltralyticsYOLOv5: V4.0.nn.SiLU(activations weights & biases) logging PyTorch hub integration (version v4.0),” Jan. 2021. [Online]. Available: https://zenodo.org/record/44181618#YFC02nHvfC0

[24] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” 2021, arXiv:2107.08430.

[25] Q. Zhao et al., “M2Det: A single-shot object detector based on multi-level feature pyramid network,” in Proc. Conf. Artif. Intell. (AAAI), 2019, pp. 1–8.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Neural Inf. Process. Syst. (NeurIPS), vol. 25, no. 6, 2017, pp. 80–90.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[28] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Neural Inf. Process. Syst. (NeurIPS), 2016, pp. 1–9.

[29] H. Jégou, “Training data-efficient image transformers & distillation speeds weights & biases logging PyTorch hub integration (version v4.0),” Jun. 2021. [Online]. Available: https://arxiv.org/abs/2004.10934

[30] J. Dai, Y. Li, K. He, and J. Sun, “YOLOX: Beyond YOLO series in 2021,” 2021, arXiv:2107.08430.

[31] Q. Zhao et al., “M2Det: A single-shot object detector based on multi-level feature pyramid network,” in Proc. Conf. Artif. Intell. (AAAI), 2019, pp. 1–8.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Neural Inf. Process. Syst. (NeurIPS), vol. 25, no. 6, 2017, pp. 80–90.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[34] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Neural Inf. Process. Syst. (NeurIPS), 2016, pp. 1–9.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.

[36] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.

[37] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.

[38] H. Tsviak, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn. (ICML), 2021, pp. 10347–10357.

[39] 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.
