Content-Augmented Feature Pyramid Network with Light Linear Transformers

Yongxiang Gu^{1,4}, Xiaolin Qin^{1,2,4,*}, Yuncong Peng^{1,4} and Lu Li³

¹ Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu, 610041, China
² Nanchang Institute of Technology, Jiangxi Nanchang, 330044, China
³ Zenseact, Gothenburg, 41756, Sweden
⁴ University of Chinese Academy of Sciences, Beijing, 100049, China
*Corresponding Author: Xiaolin Qin
{guyongxiang19@mails.ucas.ac.cn, qinxl2001@126.com, y-c.peng@qq.com, explore.leo@gmail.com}

Abstract: Recently, plenty of work has tried to introduce transformers into computer vision tasks, with good results. Unlike classic convolution networks, which extract features within a local receptive field, transformers can adaptively aggregate similar features from a global view using self-attention mechanism. For object detection, Feature Pyramid Network (FPN) proposes feature interaction across layers and proves its extremely importance. However, its interaction is still in a local manner, which leaves a lot of room for improvement. Since transformer was originally designed for NLP tasks, adapting processing subject directly from text to image will cause unaffordable computation and space overhead. In this paper, we utilize a linearized attention function to overcome above problems and build a novel architecture, named Content-Augmented Feature Pyramid Network (CA-FPN), which proposes a global content extraction module and deeply combines with FPN through light linear transformers. What's more, light transformers can further make the application of multi-head attention mechanism easier. Most importantly, our CA-FPN can be readily plugged into existing FPNbased models. Extensive experiments on the challenging COCO object detection dataset demonstrated that our CA-FPN significantly outperforms competitive baselines without bells and whistles. Code will be made publicly available.

Keywords: Object detection; feature pyramid network; transformer; self-attention; feature interaction

1 Introduction

Since AlexNet [1] was proposed in 2012, deep learning represented by convolutional neural networks (CNNs) has fundamentally changed the processing approaches in computer vision tasks over the past few years. The performance of image classification [2], object detection [3], and semantic segmentation [4] etc. have got a huge boost, which demonstrated the powerful capacity of deep CNNs, especially the incredible fitting and representation capacity. For ImageNet [5] classification, the top-1 error and top-5 error has dropped to 14.3% and 2.4% [6], respectively, which outperformed human level. However, the performance of object detection still has great room for improvement since the box mAP of the best performing detector on COCO test-dev is 58.7% [7].

Different from classification task, object detection needs to localize the object instance additionally, which requires focusing on not only the most discriminative but also other relevant parts. The former relies on high-level semantic information, while the latter needs to pay more attention to low-level localization information. Thanks to FPN [18], which achieves a good tradeoff by introducing a simple feature interaction across layers. Although there are a series of FPNs [3, 19-20] proposed to make continuous improvement, there are still some inherent defects in their common interaction mode.



Figure 1: Global view of human vision in object detection. To distinguish the indistinct "people" instance in the solid box, other similar and clear instances in the dashed boxes can be used to enhance our judgment. Therefore, it's natural to introduce query matching mechanism into feature extraction. Meanwhile, overall scene information (content in big dotted box on the left) is also crucial because the cyclist is more likely to occur in the urban road than other scenes, e.g., forest and desert.

Firstly, the interaction mode of FPN is lossy. In order to integrate features of two layers, FPN needs to align two feature maps of different sizes by interpolation or learnable convolution (e.g., transposed convolution), which will cause information loss. What's worse is that the way of interpolation may bring new issues like alias effect, while learnable convolution will increase non-ignorable computation overhead. The emergence of dilated convolution alleviates this problem to some extent.

However, another unavoidable problem is that the interaction mode of FPN is local. On one hand, backbone consists of convolutional layers performs feature aggregation by a fixed kernel from a set of nearby locations, which determines the locality of a single feature map. On the other hand, FPN simply conducts point-wise addition or learnable 1×1 convolution operation. As a result, although the receptive fields of two feature maps are variant, this feature interaction mode cannot surpass the larger receptive field. Assume that there are multiple objects of the same type at different locations (Fig. 1). When people are uncertain about one object, they will combine the background and search for similar objects to make synthetic judgment. Unfortunately, vanilla FPN lacks this capability due to its inherent network structure.

From the above considerations, the core idea of this paper can be expressed as follows:

$$I_{fine} = f_{agg} \left(f_{search} \left(I_{corase} \right), f_{background} \left(I_{corase} \right), I_{corase} \right) \tag{1}$$

where I_{corase} and I_{fine} are the corase-/fine-grained instances, $f_{search}(\cdot)$ is the query matching function for searching similar instances, $f_{background}(\cdot)$ is the global content extraction function for learning targeted contextual information, $f_{agg}(\cdot)$ is the aggregating function.

Inspired by recent works [21-24] about introducing transformers into computer vision. We note that transformer can model global relationships by self-attention mechanism, which is the key to overcome above inherent defects in FPN. We will discuss the mechanism of transformer detailly in Section 2. Since transformer was originally designed for NLP tasks, adapting processing subject directly from text to image will cause unaffordable computation and space overhead. For text data, the number of queries depends on the length of the text, while for images it is the number of pixels. Intuitively, the former is on the order of ten and the latter is on the order of ten thousand.

In this work, we overcome above problems by introducing a linearized attention function. Further, we build a novel architecture, named CA-FPN, which proposes a global content extraction module to simulate global view of human vision in object detection and deeply combines with FPN through light linear transformers. Benefit from the low complexity of light transformers, we augment each feature map in FPN with the global content extraction module respectively. As a result, objects of different sizes can learn targeted contextual information.

Faster RCNN [11] is a classic two-stage detector and has become the de facto standard architecture, which integrates feature extraction, proposal extraction, bounding box regression and classification into one network, greatly improves the comprehensive performance. What's more, Faster RCNN is well-established and highly-optimized on detectron2 [28]. Therefore, our experiments are carried out on this basis.

We highlight our principal contributions as follows:

- 1) To simulate global view of human vision in object detection, we propose a global content extraction module and equip it on top of FPN.
- 2) To address inherent defects of interaction mode in FPN, we deeply combine our global content extraction module with FPN through transformers.
- 3) Instead of introducing vanilla transformer in NLP, we utilize linearized attention function to reduce the computation and space overhead. Further, light transformers can make the application of multi-head attention mechanism easier.

The rest of this paper is organized as follows. In Section 2, we first briefly review the mainstream object detection framework and then systemically introduce FPN and transformer. Some related techniques, which share the similarity with CA-FPN, will be center on in this Section. Our proposed CA-FPN is presented detailly in Section 3. In Section 4, we report the experimental results and analyses. Finally, we conclude this paper in Section 5.

2 Related Work

2.1 Object Detection

Look back to modern object detection systems, its frameworks can be divided into two categories: two-stage [8-12] and one-stage [13-17]. The major difference between two frameworks in structure is that the two-stage uses the sub network of region proposals (RPN) to assist in generating proposals, while the one-stage generates proposals directly on the feature map. A modern detector is usually composed of following parts: a backbone which outputs the feature map of the whole picture, a neck or named FPN which fuses the feature maps of different scales to obtain multi-scale features (optional), and a head which is used to predict classes and bounding boxes of objects based on the proposals. In practice, a backbone is commonly pre-trained on ImageNet and a neck is usually composed of several bottom-up paths and several top-down paths.

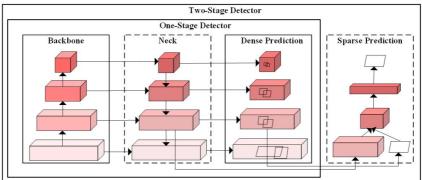


Figure 2: Structures of two-stage and one-stage detectors

Some frontier features are of great benefit to the improvement of object detection. For instance, the model scaling approach introduced in EfficientDet [3], stronger backbones e.g., CSPNet [28], and advanced activation functions e.g., Mish [30]. However, focuses on enhancing global background information and addressing inherent defects of interaction mode in FPN are neglected. Towards this end, we propose a global content extraction module and deeply combines with FPN through light linear transformers in this paper.

2.2 Feature Pyramid Network

FPN directly expands the multi-scale structure in image pyramid to feature dimension. On one hand, high-level features own low resolution and high semantic information, which contribute to classification instead of localization. On the other hand, low-level features own high resolution and low semantic information, which contribute to localization instead of classification. Therefore, both localization information and semantic information should be considered in cross-scale feature fusion. Since a CNN naturally forms a feature pyramid through its forward propagation, FPN shows great advances for detecting objects with a wide variety of scales. Currently, FPN has become a basic building block of modern object detection framework [31]. In vanilla FPN [18], a top-down pathway with lateral connection is introduced to combine multi-scale features. Then, predictions are made independently at all levels with corresponding appropriate proposals. Fig. 3 presents the architecture of FPN.

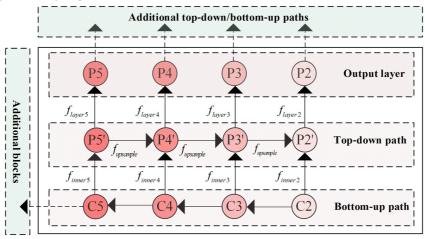


Figure 3: Architecture of FPN. The red parts are basic blocks in vanilla FPN [18] and the green parts are optional blocks to enhance feature representation. Additional blocks are in charge of increasing the receptive field while additional top-down/bottom-up paths are in charging of frequent feature interactions.

FPN integrates features of two layers in the following manner:

$$P_{i} = f_{layer_{i}} \left(f_{inner_{i}} \left(C_{i} \right) + f_{upsample} \left(P'_{i+1} \right) \right) \tag{2}$$

where C_i is the output of i-th stage's last residual block in ResNet [35], P_i' is the feature map after adjusting the channel, P_i is the output feature map for classification and regression, $f_{upsample}(\cdot)$ is the nearest neighbor upsampling function for matching the size of two feature maps, $f_{inner}(\cdot)$ is a 1×1 convolution operation for reducing the channel dimension, $f_{layer}(\cdot)$ is a 3×3 convolution operation for alleviating the aliasing effect of upsampling.

In recent years, a multitude of FPN structures have been proposed [3,19-20]. In PANet [19], an extra bottom-up path augmentation (BPA) is added to vanilla FPN for boosting information flow. Additionally, adaptive feature pooling is introduced to aggregate features usefully. In Libra R-CNN [41], the deep integrated and balanced semantic feature is utilized to strengthen the multi-level features in FPN. Besides hand-crafted designing, NAS-FPN [20] uses neural architecture search (NAS) with reinforcement learning to automatically design the optimal FPN structure in the given search space. Although it has achieved better performance, the search process is extremely time-consuming and information flow is difficult to interpret. Further, EfficientDet [3] proposes Bi-FPN, which simplifies nodes that contribute little to feature fusion and adds cross layer connection between nodes based on PANet. This direct way improves the efficiency of flowing information. In addition, EfficientDet introduces a fast fusion approach and expanded the model scaling approach introduced in EfficientNets [32], which proposes a compound scaling approach that uniformly scales the resolution, depth, and width for all backbone, neck, and head in object detection framework.

Apart from the expansion in structure and depth, additional blocks are popular to be equipped on top of FPN to increase the receptive field [16]. Although deep convolutional neural network has been able to obtain a large enough receptive field in theory, some works [36,37] indicate that the effective receptive field size increases linearly with the square root of the network's depth, therefore, at a much slower rate than what we would expect. Generally, parallel or stacking pooling and dilated convolution operations are utilized to obtain rich features of multiple receptive fields, SPP [38], ASPP [39] and DenseASPP [40] are representative modules. Although these modules share similar ideas with simulating global view of human vision, their interaction mode is plain that objects of different sizes cannot learn targeted contextual information. AC-FPN [42] designs content extraction module and attention modules for feature interaction, but their content extraction module is merely combined with the topmost feature map deeply.

In general, the design of FPN is evolving towards high efficiency and hierarchical stacking. Although frequent feature interactions can promote the generated feature map to obtain multi-scale features, this enhancement is limited by the interaction mode unless introducing global interaction mechanism.

2.3 Transformer

As is known to all, transformer was introduced by Vaswani et al. [33], which is a deep neural network based solely on attention mechanism, and has achieved impressive results on a variety of NLP tasks. It is well known for its self-attention mechanism and multi-head attention mechanism. Transformer blocks are characterized by self-attention mechanism, multi-head attention mechanism, position-wise feed-forward network, layer normalization modules and residual connections [33]. Among them, multi-head attention mechanism is an extension case of self-attention mechanism. Non-local Network [27] presents a unique deduction of self-attention in the field of computer vision, treating self-attention as the development of BM3D [34] in feature dimension. Anyhow, the generalized form of transformer operation can be formulated as:

$$q_i^m = f_q(x_i^m), \quad k_i^n = f_k(x_i^n), \quad v_i^n = f_v(x_i^n)$$
 (3)

$$S_{i,j}^{m,n} = F_{sim}\left(q_i^m, k_j^n\right) \tag{4}$$

$$w_{i,j}^{m,n} = F_{norm}(s_{i,j}^{m,n})$$
 (5)

$$\tilde{\mathbf{x}}_{i}^{m} = F_{mul}\left(\mathbf{w}_{i,j}^{m,n}, \mathbf{v}_{j}^{n}\right) \tag{6}$$

$$y_i^m = F_\theta\left(\tilde{x}_i^m\right) \tag{7}$$

where x_i^m is a feature vector at i-th position in m-th feature map $X^m \in R^{C_m \times H_m \times W_m}$, $f_q(\cdot)$, $f_k(\cdot)$, $f_v(\cdot)$ are the linear transformation functions for feature embeddings, q, k, v vectors are usually called query, key and value respectively. We denote the feature map of q, k, v vectors as Q, K, and V. $F_{sim}(\cdot)$ is the similarity function (default as dot product), $F_{norm}(\cdot)$ is the normalizing function (default as SoftMax), $F_{mul}(\cdot)$ is the weight aggregation function (default as matrix multiplication), $F_{\theta}(\cdot)$ is the non-linear transformation functions for fine-grained feature extraction.

It should be noted that $F_{\theta}(\cdot)$ is separately called position-wise feed-forward network in [33] and becomes one inseparable part of self-attention operation nowadays [22]. When m equals n in Eq. (3), it means that the query map is as same as the queried map, the above operation is worthy of the name "self-attention". While applying self-attention mechanism to fuse features between two different feature maps in this paper, it's more practical to call it as interactive attention.

Self-attention can establish global dependencies by calculating the similarity of all locations to the query point. This means that not only the weights used to aggregate features can adapt to the content, but also the receptive field is global and dynamic. However, this is not without shortcomings. The computational and space complexity of standard self-attention (SA) module are shown in follows [7]:

$$\Omega_T(SA) = 4HWC^2 + 2H^2W^2C \tag{8}$$

$$\Omega_{\rm s}(SA) = H^2 W^2 \tag{9}$$

where both computation and space complexity are quadratic to image size (we omit SoftMax computation in determining complexity here). Since the shape of feature map is usually on the order of hundreds, standard self-attention is extremely time and space consuming.

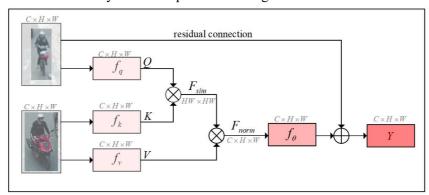


Figure 4: Architecture of self-attention operation.

When linearly projecting the query, key and value repeatedly with different and learnable linear projections, we can get different responses. Multi-head attention is the process of aggregating different responses. We begin by abstracting Eq. (4-7) and linear projecting into one function:

$$y_i^{m,h} = f_{self}\left(f_q^{head_h}\left(q_i^m\right), f_k^{head_h}\left(k_j^n\right), f_v^{head_h}\left(v_j^n\right)\right) \tag{10}$$

Then the form of multi-head attention can be expressed as follows:

$$\tilde{y}_i^m = f_{cn} \left(f_{concat} \left(y_i^{m,1}, \dots, y_i^{m,h} \right) \right) \tag{11}$$

where f_{self} is the abstract function for Eq. (4-7), $f_q^{head_h}(\cdot)$, $f_v^{head_h}(\cdot)$, $f_k^{head_h}(\cdot)$ are linear projecting functions for q, k, v in h-th head, $f_{concat}(\cdot)$ is the concatenation function in channel dimension, $f_{cp}(\cdot)$ is the compression function (default as 1×1 convolution operation) for reducing the channel dimension.

Since there exists no recurrence and no convolution in vanilla transformer, extra positional encoding is introduced. While CNNs employ inherent positional encoding, the combination of transformers with FPN requires no additional consideration of positional encoding. This article will not go into details, please refer to [33] for more learnings about positional encoding.

Through the above introduction, transformer is an ideal approach to address inherent defects of interaction mode in FPN intuitively. As far as we know, FPT [21] makes an early attempt to combine FPN with transformer but uses a more complex similarity function than vanilla transformer. Swin transformer [7] is an outstanding work, which achieves SOTA performance in several computer vision tasks, simplifying model complexity by limiting the field of view to the fixed and using shift windows to obtain dynamic and larger receptive field. In this paper, we overcome model complexity problem with light linear transformers and preserve the global receptive field. It is noted that our linear transformer is quite different from swin transformer and thus we can learn from each other in future work.

3 Content-Augmented Feature Pyramid Network

Built upon the vanilla FPN [18], our proposed model has two novel components: 1) Global Content Extraction Module (GCEM) 2) Linear Transformer (LT). The former is designed for simulating global view of human vision while the latter is adapted to address inherent defects of interaction mode in FPN.

3.1 Global Content Extraction Module

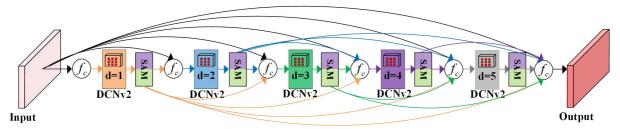


Figure 5: Architecture of GCEM

Context information is crucial for improving the final results of object detection, because it can provide additional relevant information and activate semantic association strategies to some extent. In semantic segmentation, DenseASPP [40] is a popular and well-designed module for enlarging receptive field and capturing multi-scale information. Therefore, we decide to design our global content extraction module based on DenseASPP. Firstly, we introduce the advanced technology of CNNs, known as deformable convolution network v2 (DCNv2) [44], into DenseASPP, which can improve ability to learn transformation-invariant features. However, such improvements are not without cost. Since offsets are introduced in the position of convolution, DCN can extract geometric semantic information while part of location information is lost in the meanwhile. This situation may be worse on account of introducing the modulation mechanism in DCNv2.

To alleviate the loss in localization information, we decide to introduce the Spatial Attention Module (SAM) in CBAM [45], to refine the features. Fig. 6 presents the architecture of SAM. Specifically, we introduce residual connection (the dotted connection in Fig. 6) into the origin SAM to facilitate the flow of information and optimize training period. We call the new module as residual SAM. Residual SAM can work as a feature selector in forward propagation, paying attention to important features and suppresses unnecessary signals. Also, Residual SAM can also work as a gradient update filter in the process of back propagation, getting robust to noise signal and noise label. Therefore, the defects of DCNV2 can be made up to some extent.

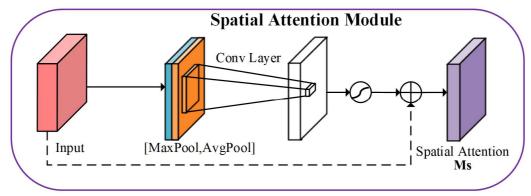


Figure 6: Architecture of residual SAM

The overall architecture of our GCEM is presented in Fig. 5. We take the topmost feature map of backbone as the input of GCEM and use the output feature map to enhance other feature maps in FPN through linear transformers. Our GCEM consists of a basic module containing the compressing function f_c , DCNv2, and residual SAM, which is stacked five times with dense connections in total. Among them, f_c first concatenates the input features, and then conduct 1×1 convolution to compress the channel number to 512. DCNv2 adjusts the channel number from 512 to 256 and SAM refines the features without changing the channel number. Although DCNv2 is an extension of dilated convolution, it can learn the shape and dilation rate dynamically. We set the dilation rates of 5 basic modules as [1,2,3,4,5] to get a default initialization.

3.2 Linear Transformer

When the dot product similarity and SoftMax are set as the similarity function and normalization function in generalized transformer, we can get the standard self-attention (SA) as follows:

$$Attention_{SA}(Q, K, V) = softmax(Q^{\top}K)V$$
(12)

$$Attention_{SA}\left(q_i^m, k_j^n, v_j^n\right) = \frac{\sum_{j=1}^N e^{q_i^{m\top} k_j^n} v_j^n}{\sum_{j=1}^N e^{q_i^{m\top} k_j^n}}$$
(13)

It is easy to find that the key factor limiting computation and space complexity is the SoftMax function. Considering that Q, K and V after reshaping have the same shape, i.e., $C \times HW$. For the first step to calculate the similarity function, computation and space complexity are $O(CH^2W^2)$ and $O(H^2W^2)$, respectively. For the second step for normalizing, computation complexity is $O(CH^2W^2)$. In general, $C \ll H \times W$ and $H \times W$ is usually on the order of ten thousand. Thus, keep the calculations in transformers and adapt processing subject directly from text to image will cause unaffordable computation and space overhead.

For successive matrix multiplication of Q^T , K, V, we can simplify operations by using the commutative law of matrix multiplication, i.e., translating $((Q^T)(K))(V)$ to $(Q^T)((K)(V))$. For the Hadamard product of K and V, computation and space complexity are $O(C^2HW)$ and $O(C^2)$, respectively. To achieve this effect, kernel function [46] is introduced to satisfy the commutativity. In this paper, we use taylor expansion to approximate SoftMax function, which can act as a kernel function and have practical interpretability. Most importantly, our linear transformer is easy to implement and facilitates extensibility.

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{14}$$

We use the first order expansion of e^x to approximate it and thus we can simplify the Eq. (13). Meanwhile, we normalize Q and K to meet the non-negative requirement of similarity function. Finally, the linear form of self-attention can be formed as follows:

$$Attention_{LT}\left(q_{i}^{m}, k_{j}^{n}, v_{j}^{n}\right) = \frac{Nv_{j}^{n} + \frac{q_{i}^{m^{\top}}}{\|q_{i}^{m^{\top}}\|} \sum_{j=1}^{N} \left(\frac{k_{j}^{n}}{\|k_{j}^{n}\|} v_{j}^{n}\right)}{N + \frac{q_{i}^{m^{\top}}}{\|q_{i}^{m^{\top}}\|} \sum_{j=1}^{N} \left(\frac{k_{j}^{n}}{\|k_{j}^{n}\|}\right)}$$
(15)

By updating the form of self-attention, the computation and space complexity are greatly simplified, especially for large feature maps. For feature maps with high-level semantic information, the features of different channels clearly correspond to different types of responses. Therefore, it is necessary to calculate attention maps according to different channels. Specifically, we first divide q_i^m and k_j^n into S equal parts. Then, we calculate the attention maps for every pair. We think of this process as the multi-head attention mechanism for computer vision. Benefit from light transformers, the application of multi-head attention mechanism can be easier. We conduct feature interaction between GCEM and feature maps in FPN via LT., The partition number is set to 1 as default and is set to 2 for feature interactions at the top two levels.

$$q_{i}^{m} = Concat(q_{i}^{m,1}, q_{i}^{m,2}, ..., q_{i}^{m,s}), \quad k_{i}^{n} = Concat(k_{i}^{n,1}, k_{i}^{n,2}, ..., k_{i}^{n,s})$$
(16)

$$Multi-head_{LT}\left(q_i^m, k_j^n, v_j^n\right) = \frac{1}{S} \sum_{s=1}^{S} Attention_{LT}\left(q_i^{m,s}, k_j^{n,s}, v_j^n\right)$$

$$\tag{17}$$

3.3 Overall Architecture

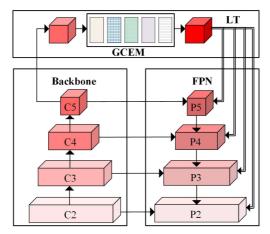


Figure 7: The overall architecture of CA-FPN

Fig. 7 presents the overall architecture of CA-FPN. Briefly, a global content extraction module (GCEM) is proposed to simulate global view of human vision in object detection and is deeply combined with FPN through linear transformers (LT). Benefit from the low complexity of light transformers, we augment each feature map in FPN with the global content extraction module respectively. It means that we take the feature map generated by GCEM as the queried map and take the feature maps in FPN as the query maps. The feature map of contextual information generated by LT will conduct point-wise addition with the corresponding input feature map in FPN. As a result, objects of different sizes can learn targeted contextual information. Although our GCEM seems complex, a good balance between calculation and performance can be got since we apply GCEM to the topmost feature map of backbone, which owns the smallest map size.

4 Experimental Design and Analysis

We show that CA-FPN significantly outperforms competitive baselines in quantitative evaluation on COCO. Some visual detection comparisons are also shown in results. Then, we provide a detailed ablation study of GCEM and LT with quantitative results.

4.1 Implementation

Experiments on object detection are conducted on MS-COCO 2017, which has 80 classes, containing 118k training images (trainval35k) and 5k validation images (minival). Following the common practice [18], backbones are first pre-trained on ImageNet [1].

Our implementation is based on detectron2 [28], with default settings outside of the FPN. For detail implementation and explanation, we highly recommend referring to the documentation in detectron2. In training stage, input images are resized such that the shorter side has 800 pixels and the longest side has 1333 pixels. Scale jittering, horizontal flipping and smooth L1 loss are used for data augmentation and regularization, respectively. Following PANet [19], we use group normalization in CA-FPN. Networks are trained on 1 NVIDIA 2080Ti GPU. We set batch-size as 2 instead of the default 16 and follow the linear scaling rule for adjusting learning rates [43]. Finally, we set the weight decay as 0.0001, momentum as 0.9, and the initial learning rate as 0.0025. In 1× training period (~12 COCO epochs), there are 720k iterations in total. The learning rate is divided by 10 at 480k and 640k. In 3× training period (~37 COCO epochs), we triple the total iteration number to 2160k and the learning rate decline moments correspondingly to 1680k and 2000k. In all training periods, we take linear warm-up strategy in the first 8k iterations.

For evaluation, we adopt the standard evaluation metrics, i.e., AP, AP_{50} , AP_{75} , AP_{8} , AP_{M} and AP_{L} . The last three measure performance with respect to objects of small, medium, and large sizes. AR, AR_{8} , AR_{M} and AR_{L} are also reported in ablation study for further analysis.

4.2 Results

We evaluate the effect of CA-FPN with different detectors (Faster R-CNN and RetinaNet). They are the representation of the two-stage and one-stage detectors, respectively. Comparisons with baselines and its competitors are reported in Tab. 1 and Tab. 2, respectively.

Table 1: Object detection comparisons using Faster R-CNN on MS-COCO 2017 minival set. * indicates that results are reported in original papers.

Faster R-CNN	backbone	period	#params	AP	AP ₅₀	AP ₇₅	AP_S	AP_{M}	AP_{L}
	ResNet-50	1×	41.7M	38.0	58.8	41.4	22.5	41.5	49.4
EDNI [19] (2017)	ResNet-50	3×	41.7M	40.0	60.6	43.6	23.9	43.3	52.2
FPN [18] (2017)	ResNet-101	1×	60.6M	40.3	61.5	43.9	24.1	44.2	51.7
	ResNet-101	3×	60.6M	42.0	62.5	45.9	25.2	45.6	54.6
BPA [19] (2018) +FPN	ResNet-50	1×	46.4M	38.6	60.0	41.8	22.8	42.1	50.1
Dense ASPP [11] (2018) +FPN	ResNet-50	1×	54.6M	39.4	61.0	42.9	23.5	42.9	51.3
Libra R-CNN [41] (2019)	ResNet-50	1×	42.0M	38.7	59.9	42.0	22.5	41.1	48.7
AC-FPN* [42] (2020)	ResNet-50	1×	54.6M	40.1	62.5	43.2	23.9	43.6	52.4
FPT [21] * (2020)	ResNet-50	1×	88.2M	38.0	57.1	38.9	20.5	38.1	55.7
CA EDN (ours)	ResNet-50	1×	55.9M	40.5	61.7	44.1	23.1	44.1	53.9
CA-FPN (ours)	ResNet-50	3×	55.9M	42.0	63.3	45.6	25.5	45.2	55.4

As one of the prevalent components, FPN is widely used in the current detection models to increase the performance of multi-scale detection. Tab. 1 shows that our method outperforms FPN by a large margin. Using ResNet-50 in 1× training period, CA-FPN improves AP, AP₅₀, AP₇₅ compared to FPN by 2.5%, 2.9%, 2.7%, respectively. Qualitative results built upon ResNet-50 in 1× training period are illustrated in Fig. 8. What's more, our CA-FPN built upon ResNet-50 even achieves comparable performance with FPN built upon ResNet-101 in the same training period. Comparing with other advanced methods, our CA-FPN is still advantageous, especially for big objects. Although FPT achieves best performance for big objects, its overall performance is poor. In contrast, our method not only performs well for big objects, but also achieves an advanced overall performance. The overall performance improvement of our method is in line with expectations. Although objects of different sizes can all gain the background augmentation and aggregate information from similar objects, large objects with more spatial locations interact more with the whole feature map.

Table 2: Object detection comparisons using RetinaNet on MS-COCO 2017 minival set

RetinaNet	+our modules	#params	AP	AP ₅₀	AP ₇₅	AP_S	AP_{M}	AP_L
baseline		37.9M	37.4	56.5	40.0	21.5	41.5	47.7
CA-FPN (ours)	✓	52.1M	38.7	58.8	41.2	23.9	43.5	47.8
im	11.3	↑2.3	11.2	↑2.4	↑2.0	↑0.1		

Since our CA-FPN can be readily plugged into existing FPN-based models, it's easy to plug CA-FPN into one-stage detectors equipped with FPN, e.g., RetinaNet. The experiments are conducted using ResNet-50 in 1× training period. In Tab. 2, it can be discovered that AP can also obtain a significant improvement, which demonstrates that our CA-FPN can help network generate feature maps with more semantical information, especially for enhancing contextual information. Different form the case in Faster R-CNN, RetinaNet equipped with CA-FPN mainly improves the detection performance of small and medium objects. We speculate that this phenomenon is due to slight differences in neck structure, neck in RetinaNet removed the P2 feature map and thus the semantic gap between the output feature map of GCEM and the underlying

feature map of neck is reduced. As a result, contextual information about small and medium objects can be improved well through LT.

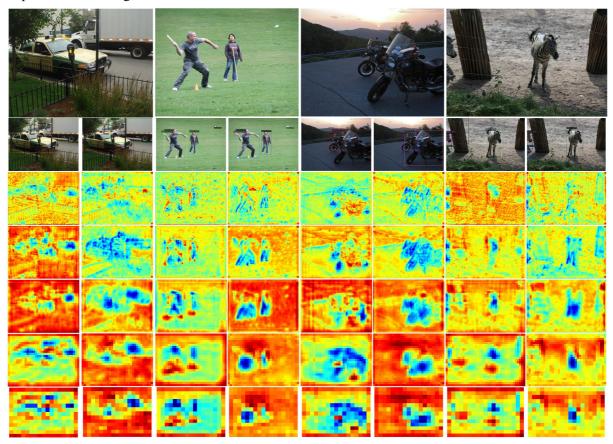


Figure 8: Comparisons of FPN and CA-FPN using Faster R-CNN on COCO test set. From top to bottom, input images, detection results and activation maps of P2-P6 are respectively shown. The left and right columns under each large input image are the results of vanilla FPN and CA-FPN respectively.

To demonstrate the feature extraction capability of CA-FPN, we randomly select images form COCO test set and feed them into detectors. The feature at the 0th channel is visualized from each output in neck. As shown in Fig. 8, the activated parts of the feature map are highlighted in blue. Take the third image for example, vanilla FPN only mainly activates the wheels, which own the most discriminant information. However, benefitting from the global awareness of GCEM and excellent feature interaction capabilities of LT, our CA-FPN can infer in conjunction with information from the larger receptive field. As a result, non-salient while relevant parts can also be activated, which will not only help to determine the category but also pinpoint the location.

In short, the experimental results demonstrate that our CA-FPN, without bells and whistles, brings consistent improvement over the baselines and is robust to various settings. Through the test on standard COCO data set, we also proved that contextual information is of great significance to objects of different sizes.

4.3 Ablation Studies

In this subsection, we perform an ablation study to evaluate the performance of GCEM and LT with quantitative results. The experiments are conducted with the Faster R-CNN. Still, the vanilla FPN is adopted as the baseline. We combine our modules gradually with the vanilla FPN and test the AP values of each combined model. When only combining with GCEM, we conduct point-wise addition between the feature

map generated by GCEM and the topmost feature map of FPN. While only combining with LT, we conduct feature interaction between the topmost feature map of FPN and other feature maps via LT. As shown in Tab.3, the model incorporated with both GCEM and LT achieves the best performance in AP, AP₅₀ and AP₇₅. It should be noted that our GCEM yields 2.2 AP improvement, which shows that the background information is crucial to enhance the overall features of objects. Moreover, our LT can further enhance objects of different sizes in a targeted way. Results in Tab. 3 demonstrate that our modules improve the performance consistently for the objects of all sizes by capturing much richer contextual information, especially for big objects. Despite the overall performance of our CA-FPN is well, small-size instances contribute least. Adding GCEM to model equipped with LT even weakens detection performance for small objects, we speculate that this phenomenon is caused by semantic gap. Since the feature map produced by GCEM is refined by multilayer network, it is difficult to interact with the underlying features through direct linear transformation in LT.

method	+GCEM	+LT	#params	AP	AP_{50}	AP_{75}	AP_S	$AP_{M} \\$	AP_{L}	AR	AR_{S}	$AR_{M} \\$	AR_{L}
FPN			41.7M	38.0	58.8	41.4	22.5	41.5	49.4	52.5	33.9	56.0	66.1
	✓		55.1M	40.2	61.3	43.5	22.9	43.6	53.4	53.7	33.5	56.9	68.7
		✓	42.5M	38.5	60.0	41.7	23.3	42.1	49.7	52.9	34.9	56.6	65.6
	✓	✓	55.9M	40.5	61.7	44.1	23.1	44.1	53.9	53.9	33.9	57.5	68.7
	improvements		↑2.5	↑2.9	12.7	↑0.6	↑2.6	14.5	11.4	↑0.0	11.5	12.6	

Table 3: Ablation study of GCEM and LT on MS-COCO 2017 minival set

5 Conclusion and Outlook

In this paper, we build a novel architecture, named CA-FPN, containing a global content extraction module, to simulate the global view of human vision in object detection. What's more, we address inherent defects of interaction mode in FPN by deeply combining our global content extraction module with FPN through light linear transformers. Benefit from the low complexity of light transformers, we augment each feature map in FPN with the global content extraction module respectively. As a result, objects of different sizes can learn targeted contextual information. Extensive experiments on the challenging COCO object detection dataset demonstrated that our CA-FPN significantly outperforms competitive baselines. Moreover, our CA-FPN can be readily plugged into existing FPN-based models and our light linear transformers can combine with the latest transformer technologies to achieve better performance and efficiency gains.

Despite the overall performance of our CA-FPN is well and the computation and space efficiency are especially improved, the overall performance can further be improved greatly with bag of freebies. The subsequent research will focus on how to improve the detecting performance of objects of different sizes in a balanced and all-sided way by better combining features of the instance with context information. In addition, more experiments will be carried on instance segmentation, semantic segmentation and etc., to further verify the generality of our CA-FPN. Last but not least, object detection is the key to improve the ability of an autonomous agent to perceive its environment so that it can (re)act. Thus, we will also focus on applying our algorithm to practical applications, such as self-driving vehicles or augmented reality and virtual reality (AR/VR), in the following work.

Funding Statement: This work was partially supported by National Academy of Science Alliance Collaborative Program (Chengdu Branch of Chinese Academy of Sciences - Chongqing Academy of Science and Technology), National Science Foundation of China (No. 61402537), Sichuan Science and Technology Program (Nos. 2019ZDZX0005, 2019ZDZX0006, 2020YFQ0056), Talent Funding Project by the Organization Department of Sichuan Provincial Party Committee, and Science and Technology Service Network Initiative (KFJ-STS-QYZD-2021-21-001).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Conf. on Neural Information Processing Systems*, Lake Tahoe, USA. pp. 1097-1105, 2012.
- [2] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 7132-7141, 2018.
- [3] M. Tan, R. Pang and Q. V. Le, "Efficientdet: scalable and efficient object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10781-10790, 2020.
- [4] J. Tan, G. Zhang, H. Deng, C. Wang, L. Lu et al., "1st place solution of lvis challenge 2020: a good box is not a guarantee of a good mask," arXiv preprint, 2020.
- [5] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li et al., "Imagenet: a large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Sparkle East, Canada. pp. 248-255, 2009.
- [6] H. Touvron, A. Vedaldi, M. Douze and H. Jegou, "Fixing the train-test resolution discrepancy: fixefficientnet," arXiv preprint, 2020.
- [7] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei et al., "Swin transformer: hierarchical vision transformer using shifted windows," arXiv preprint, 2021.
- [8] J. Dai, Y. Li, K. He and J. Sun, "R-fcn: object detection via region-based fully convolutional networks," in *Conf. on Neural Information Processing Systems*, Barcelona, Spain, pp. 379-387, 2016.
- [9] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Columbus, Ohio, USA, pp. 580-587, 2014.
- [10] R. Girshick, "Fast r-cnn," in IEEE Int. Conf. on Computer Vision, Santiago, Chile, pp. 1440-1448, 2015.
- [11] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Conf. on Neural Information Processing Systems*, Montreal, Canada, pp. 91-99, 2015.
- [12] K. He, G. Gkioxari, P. Dollar and R. B. Girshick, "Mask r-cnn," in *IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 2961-2969, 2017.
- [13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 779–788, 2016.
- [14] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, USA, pp. 7263–7271, 2017.
- [15] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," arXiv preprint, 2018.
- [16] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOv4: optimal speed and accuracy of object detection," arXiv preprint, 2020.
- [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed, "SSD: single shot multibox detector," in *Proc. European Conf. on Computer Vision*, Amsterdam, Netherlands, pp. 21-37, 2016.
- [18] T. Y. Lin, P. Dollár, R. Girshick and K. He, "Feature pyramid networks for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Hawaii, USA, pp. 2117-2125, 2017.
- [19] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 8759-8768, 2018.
- [20] G. Ghiasi, T. Y. Lin and Q. V. Le, "Nas-fpn: learning scalable feature pyramid architecture for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 7036-7045, 2019.
- [21] D. Zhang, H. Zhang, J. Tang, M. Wang, X. Hua et al., "Feature pyramid transformer," in *Proc. European Conf. on Computer Vision*, Springer, Cham, 2020: 323-339.
- [22] H. Zhao, J. Jia and V. Koltun, "Exploring self-attention for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2020: 10076-10085.
- [23] H. Hu, Z. Zhang, Z. Xie and S. Lin, "Local relation networks for image recognition," in *IEEE Int. Conf. on Computer Vision*, pp. 3464-3473, 2019.

- [24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov et al., "End-to-end object detection with transformers," in *Proc. European Conf. on Computer Vision*, Springer, Cham, pp. 213-229, 2020.
- [25] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei et al., "Cenet: Criss-cross attention for semantic segmentation," in *IEEE Int. Conf. on Computer Vision*, pp. 603-612, 2019.
- [26] L. Huang, Y. Yuan, J. Guo, C. Zhang, X. Chen et al., "Interlaced sparse self-attention for semantic segmentation," arXiv preprint, 2019.
- [27] X. Wang, R. Girshick, A. Gupta and K. He, "Non-local neural networks," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 7794-7803, 2018.
- [28] https://github.com/facebookresearch/detectron2
- [29] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh et al., "CSPNet: a new backbone that can enhance learning capability of cnn," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 390-391, 2020.
- [30] D. Misra, "Mish: a self regularized non-monotonic neural activation function," arXiv preprint, 2019.
- [31] Z. Zou, Z. Shi, Y. Guo and J. Ye, "Object detection in 20 years: a survey," arXiv preprint, 2019.
- [32] M. Tan and Q. V. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," arXiv preprint, 2019
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones et al., "Attention is all you need," in *Conf. on Neural Information Processing Systems*, Long Beach, California, USA, pp. 6000–6010, 2017
- [34] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080-2095, 2007.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, pp. 770-778, 2016.
- [36] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang et al., "Deformable convolutional networks," in *IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 764-773, 2017.
- [37] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Conf. on Neural Information Processing Systems*, Barcelona, Spain, pp. 4905-4913, 2016.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 2015.
- [39] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2017.
- [40] M. Yang, K. Yu, C. Zhang, Z. Li and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 3684-3692, 2018.
- [41] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang et al., "Libra r-cnn: Towards balanced learning for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 821-830, 2019.
- [42] J. Cao, Q. Chen, J. Guo and R. Shi, "Attention-guided context feature pyramid network for object detection," arXiv preprint, 2020.
- [43] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski et al., "Accurate, large minibatch sgd: training imagenet in 1 hour," arXiv preprint, 2017.
- [44] X. Zhu, H. Hu, S. Lin and J. Dai, "Deformable convnets v2: more deformable, better results," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 9308-9316, 2019.
- [45] S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "Cbam: convolutional block attention module," in *Proc. European Conf. on Computer Vision*, Munich, Germany, pp. 3-19, 2018.
- [46] Y. Tay, M. Dehghani, D. Bahri and D. Metzler, "Efficient transformers: a survey," arXiv preprint, 2020.