Neural Processing Letters (2022) 54:581–595
https://doi.org/10.1007/s11063-021-10645-0

Scale-Insensitive Object Detection via Attention Feature Pyramid Transformer Network

Lingling Li[1] · Changwen Zheng[2] · Cunli Mao[3] · Haibo Deng[4] · Taisong Jin[5]

Accepted: 9 September 2021 / Published online: 19 October 2021

Abstract
With the progress of deep learning, object detection has attracted great attention in computer vision community. For object detection task, one key challenge is that object scale usually varies in a large range, which may make the existing detectors fail in real applications. To address this problem, we propose a novel end-to-end Attention Feature Pyramid Transformer Network framework to learn the object detectors with multi-scale feature maps via a transformer encoder-decoder fashion. AFPN learns to aggregate pyramid feature maps with attention mechanisms. Specifically, transformer-based attention blocks are used to scan through each spatial location of feature maps in the same pyramid layers and update it by aggregating information from deep to shadow layers. Furthermore, inter-level feature **aggregation** and intra-level information **attention** are repeated to encode multi-scale and self-attention feature representation. The extensive experiments on challenging MS COCO object detection dataset demonstrate that the proposed AFPN outperforms its baseline methods, *i.e.*, DETR and Faster R-CNN methods, and achieves the state-of-the-art results.

Keywords Object detection · Feature pyramid · Attention · Convolutional network

1 Introduction

Object detection is a fundamental problem of computer vision community, which aims at predicting a set of class labels and bounding boxes for all instances of interest in images. As an important computer vision task, object detection is the basis of many tasks, such as object tracking [1], image retrieval [2,3], image ranking [4] and instance segmentation [5], *etc*.

✉ Cunli Mao
maocunli@163.com

[1] School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450046, China

[2] Institute of Software, Chinese Academy of Sciences, Beijing 100190, China

[3] Yunnan Key Laboratory of Artificial Intelligence, Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

[4] Beijing Zhonghangzhi Technology Co.,Ltd., Beijing 100176, China

[5] School of Informatics, Xiamen University, Xiamen 361005, China

From the application view [6], object detection is divided into two research topics "detection applications" and "general object detection", where the former one aims to detect object under particular application scenarios, such as text detection, face detection, pedestrian detection, *etc*., and the latter one refers to detect different types of objects to simulate the human cognition and vision under a unified framework. Recently, convolutional neural networks (CNNs) have achieved remarkable success in object detection, which pushes it to a hot-spot research topic with unprecedented attention and leads to enormous breakthroughs. Object detection has been broadly employed in many real-life applications, *e.g*., video surveillance, robot vision, autonomous driving, *etc*.

Most existing methods formulate object detection as classification and regression problems on a large set of proposals [7] and anchors [8,9]. With such formulation, they need to introduce a series of designs to derive the final detection results, *i.e*., the near-duplicate prediction removal, the distribution of anchors and the heuristics of assigning labels. Above designs also significantly influence the detectors in term of run-time and accuracy performance. To simplify the pipeline of object detection, DETR [10] is recently proposed to leverage a direct set prediction approach for object detection by streamlining the testing and training pipeline, which achieves the competitive performance compared to the baselines. Based on the popular architecture for sequence prediction, *i.e*., transformers, DETR has an encoder-decoder structure to model the interaction of all activation pairwise in feature maps explicitly.

However, one vital challenge in the above set prediction method lies in handling scale variation. Commonly, object scale varies in a broad range, which hinders the detection ability of small and large instance especially. For example, DETR cannot obtain promising performance for small objects. To relieve the scale variation problem, one intuitive solution is to employ an image pyramid, which is popular in many deep CNN-based methods [8,11]. Particularly, most CNN-based detectors benefit from multi-scale testing and training. SNIP [12,13] proposes a scale normalization method to train the appropriate-size objects selectively in each scale, which avoids training extreme-scale objects. However, image pyramid methods increase the inference and training time, which limits the practical applications. To reduce computation cost, the other methods leverage in-network feature pyramids to approximate image pyramids. For instance, SSD [14] uses the feature maps of different layers to detects objects. FPN [15] constructs a fast feature pyramid by connecting feature maps of nearby scales. However, few works exploit feature pyramid for attention mechanisms in set-prediction-based detectors.

In this article, we propose a novel end-to-end framework, termed Attention Feature Pyramid Transformer Network (AFPTN), to learn the object detectors with pyramid feature maps via transformer encoder-decoder fashion. AFPN learns to aggregate the pyramid feature maps with attention mechanisms. In particular, attention blocks, *i.e*., transformers, are used to scan through each spatial location of feature maps and update it by aggregating information from the deep to shadow layers. AFPTN has the following two advantages: (1) Transformers are performed to attend information from all spatial locations, which are aggregated with multi-scale features in an end-to-end framework. (2) Instead of directly feeding feature pyramid to the encoder-decoder transformer, which is computationally infeasible, AFPTN repeats intra-level information **attention** and inter-level feature **aggregation** in an iteration approach by encoding multi-scale and self-attention feature representation for sequential modules.

The contributions of our work are summarized as follows:

- We propose a novel end-to-end framework, coined as Attention Feature Pyramid Transformer Network (AFPTN), to learn the object detectors with pyramid feature maps via transformer encoder-decoder fashion.

- With feasible computation cost, intra-level information attention and inter-level feature aggregation are applied to encode multi-scale, self-attention feature representation.
- The extensive experiments conducted on challenging MS COCO benchmarks show that the proposed AFPTN outperforms its baselines and achieves the state-of-the-art results.

2 Related Work

2.1 Object Detection

Recently, the CNN-based object detection methods have shown remarkable improvements in both computing speed and accuracy. As one of the predominant methods, two-stage detection paradigm [8] first predicts a set of object proposals and then refine them for final classification and regression. R-CNN [16] generates object proposals by Selective Search [7] and then regresses and classifies the object proposals sequentially and independently. To decrease the redundant computation of extracting proposal feature in R-CNN, Fast R-CNN [11] and SPPNet [17] extract the full-image feature maps and then generate proposal features through RoIPooling layer and spatial pyramid pooling layer, respectively. RoIAlign layer [5] improves RoIPooling layer by addressing the problem of coarse spatial quantization. A unified end-to-end framework for object detection is proposed by Faster R-CNN [8], which replaces the original time-consuming object proposal modules with an object proposal network that shares the same backbone network with the detection network. R-FCN [18] further improves the efficiency of Faster R-CNN by constructing a position-sensitive score maps via fully convolutional networks, which avoids the RoI-wise head. Online hard example mining [19] handles the category imbalance, which makes easy negatives overwhelm the loss and computed gradients. To be sequentially more selective against close false positives, Cascade R-CNN [20] trains a sequence of models with increasing IoU thresholds. Relation network [21] proposes to use attention module to model object relations by simultaneous interaction between geometry and appearance feature. Deformable convolutional networks [22] enhances the transformation modelling capability of detectors by augmenting the spatial sampling locations of convolution and RoIPooling layers with new offsets. Some work focuses on improving IoU metric [23], region anchors [24], sample selection [25], non-maximum suppression (NMS) [26] and noise tolerant [27]. Multi-region [28], spatial transform [29], semantic segmentation [5,30] and generative adversarial learning [31] are leveraged to boost detection performance. Anchor free detectors [32] directly find objects without preset anchors.

One-stage paradigm that is popularized by SSD [14] and YOLO [9], directly classifies pre-defined anchors without the object proposal generation step and further refines them. DSSD [33] introduces new contextual information based on the multi-layer prediction in SSD with deconvolutional layers to improve the performance. To address the huge foreground-background category imbalance that stands outs as a central issue in one-stage paradigm, RetinaNet [34] proposes focal loss. RefineDet [35] proposes an anchor refinement module to coarsely adjust the anchor boxes and filter the negative anchors for the detection module, as inherited by the merits of two-stage paradigm. Light-Head R-CNN [36] uses cheap R-CNN subnet and thin feature maps to improve the efficient of two-stages detector. DeNet [37] employs a sparse distribution estimation scheme through an end-to-end CNN-based detection model. R-FCN-3000 [38] decouples object detection and classification in real-time object detector, which multiplies the object score with the fine-grained classification score to obtain the detection score. Pelee [39] and ThunderNet [40] construct lightweight models for mobile platforms with limited computing power and memory resource.

2.2 Multi-Scale Features

To improve the accuracy of detectors on detecting difficult objects with extreme size, various strategies [12,13,15,41–43] have been proposed to introduce multi-scale information to the conventional detection framework. Image pyramids [22,44] is a common strategy to improve the performance of detectors, which detects object across scales during training and testing to remedy the scale-variation problem. However, image pyramid method increases the inference time and neglects the in-network feature hierarchy to handle large scale variation. During multi-scale training, for each resolution of input images, SNIP [12] proposes a scale normalization strategy based on the image pyramid scheme to train instances that fall into the desired scale range. SNIPER [13] samples background proposals in different scales and only selects context ones around the ground-truth bounding boxes, which performs multi-scale training more efficiently. However, SNIPER and SNIP still suffer from the unavoidable increasing of inference time.

Another stream of utilizing multi-scale information in fully supervised learning is to consider both high-level and low-level information. R-SSD [42] and RRC [43] gather both low-level and high-level feature maps by concatenation, which cost more computational resource significantly. To generate the better feature maps for prediction, ION [45] and HyperNet [46] concatenate high-level and low-level features of various layers. Before fusing multi-level features, transformation operators or specific normalization need to be developed, as the features of different layers usually have different sizes. Instead, object detection at multiple layers without feature fusion is performed in MS-CNN [47] and SSD [14].

2.3 Feature Pyramid

The feature pyramid structure has been applied to many computer vision tasks successfully, *i.e.*, semantic segmentation [41], object detection [15,44], human pose estimation [48]. The feature pyramid structure contains two steps: Firstly, an encoder captures high-level semantic information and reduces the resolution of feature maps gradually. Secondly, a decoder gradually restores the spatial cues. FPN [15] is one of the representative model architectures to generate pyramidal feature representations for object detection, which boosts the low-level feature semantic representation at bottom layers by introducing lateral connections and top-down pathway. PANet [44] proposes adaptive feature pooling to aggregate features from all levels and introduces additional bottom-up path augmentation in FPN to boost the feature hierarchies for the better performance. DSSD [33] aggregates context and enhances the high-level semantics for shallow-layer features by using deconvolution layers as decoders. U-Net [41] is an encoder-decoder model, which shares the information learned by the encoder with the decoder through concatenation with skip connections. TripleNet [49] simultaneously predicts the objects and parses pixel semantic labels by all different layers in the decoder.

Recently, the pioneering work ViT [50] demonstrates that pure Transformer-based architectures also achieves very competitive results, indicating the potential of handling the vision tasks and natural language processing (NLP) tasks under a unified framework. Built upon the success of ViT, many efforts have been devoted to designing better Transformer based architectures for various vision tasks, including low-level image processing [51], object detection [10]. The hierarchical Transformer architecture but adopt different self-attention mechanisms

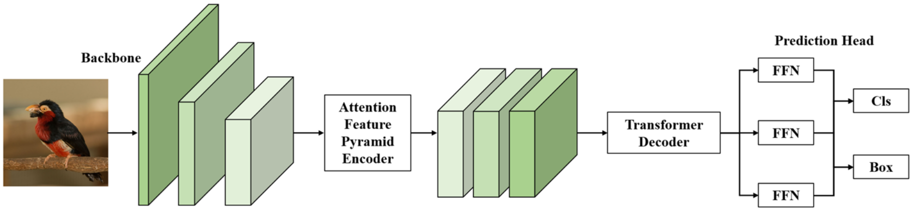

Fig. 1 The overall flowchart of the proposed attention feature pyramid network for object detection. It consists of four components: backbone CNNs, attention feature pyramid encoders, transformer decoders and prediction head. *FFN* denotes a feed forward network that generates the final detection outputs. And *Cls* and *Box* denote the classification and bounding-box regression predictions, respectively

can utilize the multi-scale features and reduce the computation complexity by progressively decreasing the number of tokens.

3 Method

3.1 The Overall Framework

We aim to leverage the pyramidal feature hierarchy and attention mechanism to formulate object detection as an end-to-end direct set prediction framework with semantic features. As illustrated in Fig. 1, our method takes an arbitrary-size single-scale image as input, and the CNN backbones output feature maps of proportional size at multiple levels. This backbone feature extraction is independent of the CNN architectures. Then we build multiple top-down pathways and combine them with attention mechanism to construct an attention feature pyramid, which is feed to the subsequent decoders for detection predictions. The rest of this section describes the details of each component.

3.2 Multi-Headed Self-Attention (MHSA)

In this subsection, we first revisit the multi-headed self-attention (MHSA) [52] structure, which is the basic module of our encoder and decoder components.

Given the key-value sequence X^{kv} of dimension $N^{\mathrm{kv}} \times d$ and the query sequence X^{q} of dimension $N^{\mathrm{q}} \times d$, the outputs of MHSA are the same dimension as the query sequence. It starts by adding the query and key positional encodings, which follows by computing query, key and value embeddings.

$$\begin{array}{l} K = T^{\mathrm{k}}(X^{\mathrm{kv}} + P^{k}) \\ V = T^{\mathrm{v}}(X^{\mathrm{kv}}) \\ Q = T^{\mathrm{q}}(X^{\mathrm{q}} + P^{\mathrm{q}}) \end{array}, \tag{1}$$

where T^{k}, T^{v} and T^{q} are weight tensors in T^{e}. The we compute the attention weights A by applying the softmax operation to dot product of query and key embeddings, which interacts all pairwise elements sequentially and explicitly.

Thus, each element in the query attends to all elements in the key-value pairs:

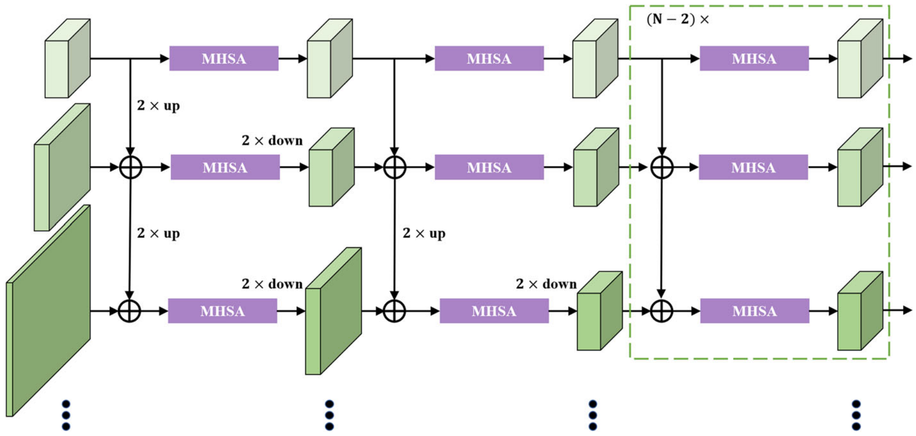

Fig. 2 The structure of the proposed attention feature pyramid encoders. Given a set of build-in feature pyramid from CNN backbones, AFP iteratively applies top-down pathways and attention mechanism to encode multi-scale, self-attention feature representation

$$\begin{aligned} A_{ij} &= \frac{1}{Z_i} e^{\frac{1}{\sqrt{d}} Q_i^T K_j} \\ Z_i &= \textstyle\sum_{j=1}^{N^{\mathrm{kv}}} e^{\frac{1}{\sqrt{d}} Q_i^T K_j} \end{aligned} . \tag{2}$$

We aggregate the values V weighted by attention weights A as the final output of single-head attention (SHA):

$$\mathrm{SHA}(\mathrm{X}^{\mathrm{q}}, \mathrm{X}^{\mathrm{kv}}, \mathrm{T})_i = \textstyle\sum_{j=1}^{N^{\mathrm{kv}}} A_{ij} V_j \, . \tag{3}$$

Then the MHA is simply the concatenation of M SHA followed by a projection layer with weight T^p:

$$\begin{aligned} \mathrm{MHA}(\mathrm{X}^{\mathrm{q}}, \mathrm{X}^{\mathrm{kv}}, \mathrm{T})_i = T^{\mathrm{p}}[\mathrm{SHA}(\mathrm{X}^{\mathrm{q}}, \mathrm{X}^{\mathrm{kv}}, \mathrm{T}_1); \ldots ; \\ \mathrm{SHA}(\mathrm{X}^{\mathrm{q}}, \mathrm{X}^{\mathrm{kv}}, \mathrm{T}_{\mathrm{M}})] \end{aligned} . \tag{4}$$

MHSA is a special case $X^{\mathrm{q}} = X^{\mathrm{kv}}$ of multi-headed attention (MHA):

$$\mathrm{MHSA}(X, T^{\mathrm{e}}, T^{\mathrm{p}}) = \mathrm{MHA}(X, X, T^{\mathrm{e}}, T^{\mathrm{p}}), \tag{5}$$

where X is the input sequence, T^e is the embedding weight, and T^p is the project weight.

3.3 Attention Feature Pyramid (AFP) Encoders

We propose a novel Attention Feature Pyramid as our encoders, which build multiple top-down pathways and combine them with attention mechanism, as illustrated in Fig. 2. In particular, we iteratively apply top-down pathways and attention mechanism to each level of build-in feature pyramid from CNN backbones. The top-down pathways combine low-level and high-level feature maps from backbones to generate strong multi-scale feature maps by a set of cross-scales connections. And attention mechanism imposes each level of feature pyramid to attend to information from different positions and representation subspaces jointly. We repeat above inter-level feature aggregation and intra-level information attention to encode multi-scale and self-attention feature representation.

We utilize FPN [15] as our basic top-down pathways structure, which is briefly revisited as below. We treat each stage from the backbone CNNs as one pyramid level. The output of the last layer of each stage is defined as our reference set of feature maps to create our pyramid, as the deepest layer of each stage has the strongest representation. The decoder upsamples spatially coarse, semantically strong feature maps from high-level pyramids to produce high-resolution feature maps. These high-resolution feature maps are then boosted with feature maps in the low-level pyramids through hidden connections. Feature maps with the same spatial size from the encoder and the decoder are merged by those hidden connections. The feature maps in encoders are accurately localized as they are only subsampled a few times, but they only have lower-level semantics.

To this end, a 1×1 convolutional layer and element-wise addition are used to align channel dimension and combine the upsampled feature maps and the corresponding bottom-up feature maps. The final feature maps have 4 levels in ResNet, corresponding to $\left\{P_1^2, P_1^3, P_1^4, P_1^5\right\}$, which are of the same spatial sizes, respectively. We also build $\left\{P_1^6\right\}$ for covering a larger scale by simply applying a subsampling on $\left\{P_1^5\right\}$ with stride 2.

With the above feature pyramid, we further employ above MHSA for each level to interact all pairwise elements, which results in feature maps of $\left\{A_1^2, A_1^3, A_1^4, A_1^5, A_1^6\right\}$. The above top-down pathways and attention mechanism are applied iteratively to get output attention maps $\left\{A_{N^{\mathrm{d}}}^2, A_{N^{\mathrm{d}}}^3, A_{N^{\mathrm{d}}}^4, A_{N^{\mathrm{d}}}^5, A_{N^{\mathrm{d}}}^6\right\}$. However, the low-level pyramids, *e.g.*, P_1^2, are high-resolution feature maps, which is inefficient to compute attention maps. Thus, we downsample the attention maps by a factor of 2 each iteration until it has the same resolution as the feature maps of higher levels. To further reduce computational cost, we merge the output feature pyramid of AFP before the decoder module. As each level of feature pyramid has the same resolution, we use element-wise addition to generate inputs for sequential modules.

3.4 Transform Decoders

The decoders have similar structure as Transform decoders in [10], each of which is the MHSA [52] to transform N embeddings of size d, as illustrated in Fig. 3. The decoder receives learnable object queries and encoder memory, and produces the output embedding. Then they are independently decoded into box coordinates and class labels by a feed forward network (FFN), resulting N final predictions. Using self- and encoder-decoder attention over these embeddings, the model globally reasons about all objects together using pair-wise relations between them, while being able to use the whole image as context. The final prediction is computed by a 3-layer perceptron with ReLU activation function and hidden dimension d, and a linear projection layer. The FFN predicts the normalized center coordinates, height and width of the box w.r.t. the input image, and the linear layer predicts the class label using a softmax function. Since we predict a fixed-size set of N bounding boxes, where N is usually much larger than the actual number of objects of interest in an image, an additional special class label is used to represent that no object is detected within a slot. This class plays a similar role to the “background” class in the standard object detection approaches.

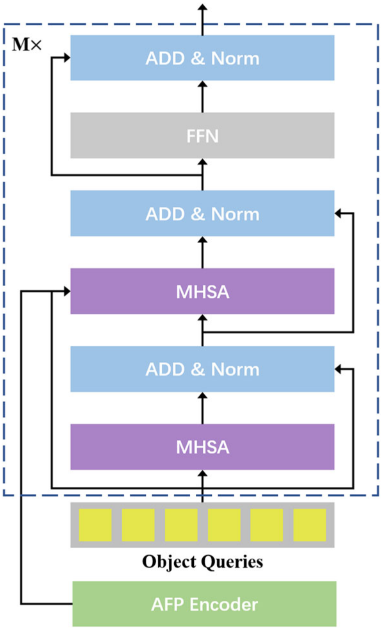

Fig. 3 The structure of transform decoders [10]. Given learnable object queries and encoder memory, decoders produces the output embedding

3.5 Optimization Objective

Our method infers a fixed-size set of N predictions as in [10]. Thus an optimal bipartite matching is used to align the prediction results to ground-truth objects, and then optimize classification loss and bounding-box regression loss. We denote the ground-truth objects as y and the predictions as $\hat{y} = \{\hat{y}_i\}_i^N$ We search the permutation of N elements σ to find a bipartite matching between the predictions and ground-truth objects.

$$\hat{\sigma} = \arg\min \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}), \tag{6}$$

where $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ is the matching cost between the ground-truth y_i and a prediction with index $\sigma(i)$. We define the predicted box as $\hat{b}_{\sigma(i)}$ and probability of category c_i as $\hat{p}_{\sigma(i)}(c_i)$ in each prediction $\hat{y}_i$.

Then matching cost is defined as:

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{I}_{\{c_i \neq 0\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{I}_{\{c_i \neq 0\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)}), \tag{7}$$

where the bounding-box loss $\mathcal{L}_{\text{box}}$ is defined as in [23]:

$$\mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\sigma(i)}\right) = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}\left(b_i, \hat{b}_{\sigma(i)}\right) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1. \tag{8}$$

With the optimal $\hat{\sigma}$, we define the Hungarian loss as optimization objective, which is similar to the losses of common object detectors:

$$\mathcal{L}_{\text{Hungarian}}\left(y_i, \hat{y}\right) = -\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{I}_{\{c_i \neq 0\}} \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\hat{\sigma}(i)}\right). \tag{9}$$

4 Experiment

We conducted the experiments on COCO. With insights and qualitative results, a detailed ablation study is provided.

4.1 Dataset

We evaluate our method on COCO 2017 [53], which contains 5k *validation* images and 118k *training* images. Each image is annotated with bounding-box labels with 7 instances for each image on average. It has up to 63 instances in images of *training* set, ranging from large to small objects.

4.2 Implementation Details

We train AFPN with AdamW [54] method and set the initial learning rate of backbone to 10^{-5}, the learning rate of the transformer to 10^{-4}, and weight decay to 10^{-4}. We use Xavier [55] to initialize all weights and leverage the ImageNet pre-trained weights for backbones with frozen batch-norm layers. We use ResNet-50 and ResNet-101 as our backbones to report the exprimental results, in terms of AFPN-R50 and AFPN-R101, respectively. As a common setting, we remove the stride from the first convolution of the last stage in the backbone and increase the feature resolution with dilation. The corresponding models are called AFPN-DC5-R50 and AFPN-DC5-R101, respectively. For scale augmentation, we resize the input images such that the longest at most $1,333$ while the shortest side is at most 800 pixels and at least 480.

During training, we also apply random crop augmentations to learn global relationships among the self-attention encoders. Particularly, each image is cropped to a random rectangular region with probability 0.5 and then resized to (800–1333). Each element i of the ground truth set can be seen as a $y_i = (c_i, b_i)$, where c_i is the target class label (which may be Ø) and $b_i \in [0, 1]^4$ is a vector that defines ground truth box center coordinates and its height and width relative to the image size. We use Hungarian algorithm to minimize the pair-wise matching cost between ground truth y_i and a prediction with index $\sigma(i)$, as described in Subsect. 3.5. This optimal assignment plays the same role as the heuristic assign- ment rules used to match proposal [8] or anchors [15] to ground truth objects in modern detectors.

The dropout ratio in the Transformer is set to 0.1. We also override the prediction of empty slots with the second-highest scoring category and the corresponding confidence to improve AP.

For ablation study, we train AFPN for 300 epochs, which drops the learning rate after 200 epochs by a factor of 10. To compare with the-state-of-the-art detectors, we train for 500 epochs, which drops the learning rate by a factor of 10 after 400 epochs.

4.3 Comparison with the Baselines

We use the attention-based and popular detectors, *i.e.*, DETR and Faster R-CNN methods, as our baselines. Our method AFPN consistently outperforms other methods for all metrics in Table 1. We observe that AFPN has a large improvement for small and medium object compared to DETR , as illustrated by AP_S and AP_M metrics.

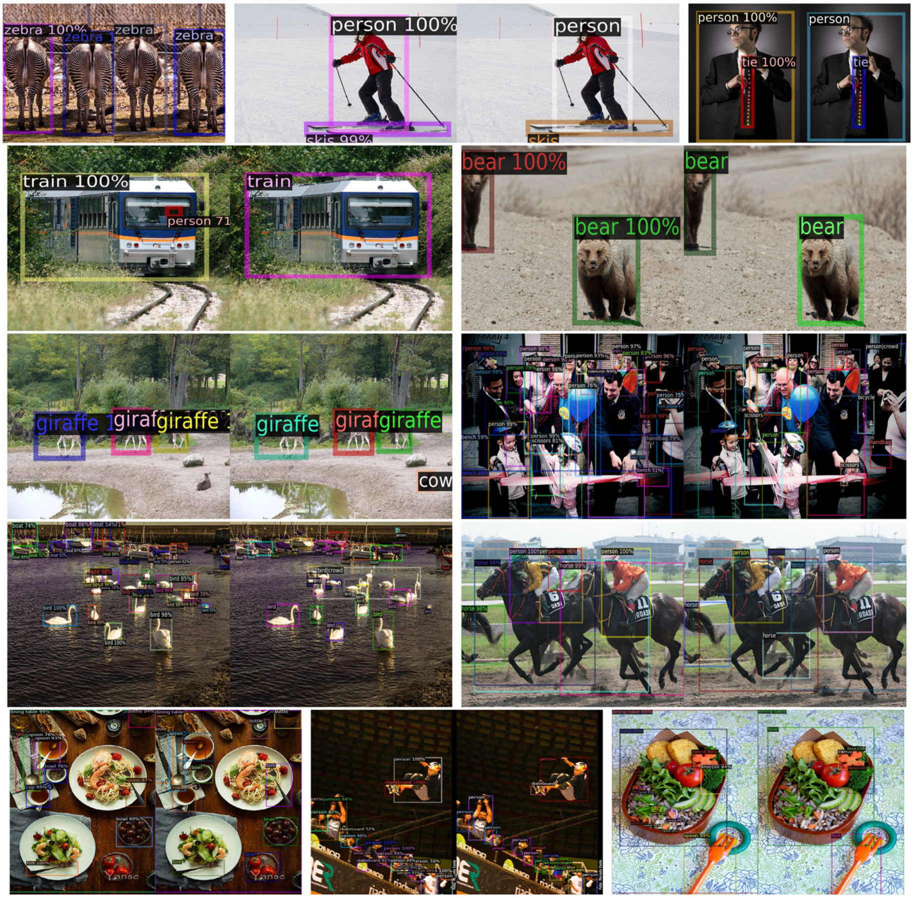

Fig. 4 Visualization results on the MS COCO 2017 *val*.AFPN outputs and ground-truth segmentation are presented from left to right in each group

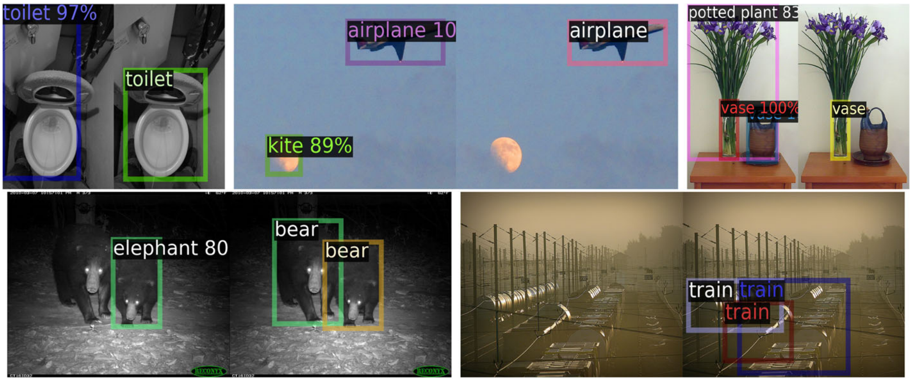

Fig. 5 Visualization failure cases on the MS COCO 2017 *val*. AFPN outputs and ground-truth segmentation are presented from left to right in each group. As shown in the last column, our failure modes mainly come from two parts: (1) confusion with similar objects, and (2) low-quality images

Table 1 Comparison AFPN with Faster R-CNN and DETR on the COCO validation set. The first section shows the results of Faster R-CNN models

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-R50-DC5	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-R50-FPN	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-R50-DC5+	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-R50-FPN+	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	44.0	63.9	47.8	27.2	48.1	56.0
DETR -R50-C5	42.0	62.4	44.2	20.5	45.8	61.1
DETR -R50-DC5	43.3	63.1	45.9	22.5	47.3	61.1
DETR -R101-C5	43.5	63.8	46.4	21.9	48.0	61.8
DETR -R101-DC5	44.9	64.7	47.7	23.7	49.5	62.3
AFPTN-R50-C5	45.1	65.2	47.1	23.6	47.9	64.0
AFPTN-R50-DC5	46.5	66.1	49.1	25.6	50.2	64.2
AFPTN-R101-C5	46.9	67.1	49.3	24.1	50.5	64.3
AFPTN-R101-DC5	**48.3**	**68.0**	**50.9**	**28.3**	**52.3**	**65.5**

The second section shows results for Faster R-CNN models with the long `9x` training schedule, random crops train-time augmentation and GIoU [23]. The third section shows the results of DETR in [10]. ResNet-50 and ResNet-101 backbones are used

Table 2 Ablation study of AFPN on different pyramid levels on COCO `val2017` with ResNet-50 backbone

A^6	A^5	A^4	A^3	A^2	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
	✓				43.3	63.1	45.9	22.5	47.3	61.1
✓	✓				44.1	64.3	47.2	23.4	48.1	62.2
✓	✓	✓			45.8	65.1	48.4	24.1	49.8	63.7
✓	✓	✓	✓		46.5	66.1	49.1	25.6	50.2	64.2
✓	✓	✓	✓	✓	46.4	66.1	49.2	25.7	50.1	64.2

4.4 Ablation Studies

Table 2 lists the ablation study of various pyramid levels, where we present the sources of their improvements. Our baseline is DETR , which only uses the feature maps from the last stage of backbones to generate attention feature maps, *i.e.*, A^5. We demonstrate that employing more feature pyramid has the consistent improvement of learning performance. The finest pyramid, *i.e.* A^2, only has margin gains, but it requires enormous computation for attention and aggregation. Thus, we only use four pyramid levels, *i.e.*, $\left\{A^3, A^4, A^5, A^6\right\}$, in other experiments.

4.5 Case Studies

The qualitative results on the MS COCO 2017 *val* are shown in Fig. 4. Our approach outputs semantically meaningful and precise predictions despite the existence of complex object appearances and challenging background contents. It demonstrated the effectiveness of the

Table 3 State-of-the-art comparison on COCO `test-dev` for bounding-box object detection

Method	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv3 [56]	DarkNet-53		33.0	57.9	34.4	18.3	25.4	41.9
RetinaNet [34]	ResNeXt-101		40.8	61.1	44.1	24.1	44.2	51.2
RefineDet [35]	ResNet-101	✓	41.8	62.9	45.7	25.6	45.1	54.1
CornerNet [57]	Hourglass-104	✓	42.1	57.8	45.3	20.8	44.8	56.7
ExtremeNet [58]	Hourglass-104	✓	43.7	60.5	47.0	24.1	46.9	57.6
FSAF [59]	ResNeXt-101	✓	44.6	65.2	48.6	29.7	47.1	54.6
FCOS [32]	ResNeXt-101		44.7	64.1	48.4	27.6	47.5	55.6
CenterNet [60]	Hourglass-104	✓	45.1	63.9	49.3	26.6	47.1	57.7
NAS-FPN [61]	AmoebaNet		48.3	–	–	–	–	–
SEPC [62]	ResNeXt-101		50.1	69.8	54.3	31.3	53.3	63.7
SpineNet [62]	SpineNet-190		52.1	71.8	56.5	35.4	55.0	63.6
Mask R-CNN [5]	ResNet-101		39.8	62.3	43.4	22.1	43.2	51.2
Libra R-CNN[63]	ResNeXt-101		43.0	64.0	47.0	25.3	45.6	54.6
PANet [44]	ResNeXt-101		47.4	67.2	51.8	30.1	51.7	60.0
DCN-v2 [64]	ResNet-101	✓	46.0	67.9	50.8	27.8	49.1	59.5
SNIP [12]	Model Ensemble	✓	48.3	69.7	53.7	31.4	51.6	60.7
SINPER [13]	ResNet-101	✓	47.6	68.5	53.4	30.9	50.6	60.7
TridentNet [65]	ResNet-101	✓	48.4	69.7	53.5	31.8	51.3	60.3
TSD [66]	SENet154	✓	51.2	71.9	56.0	33.8	54.8	64.2
MegDet [67]	Model Ensemble	✓	52.5	-	-	-	-	-
CBNet [68]	ResNeXt-152	✓	53.3	71.9	58.5	35.5	55.8	66.7
HTC [69]	ResNeXt-101		47.1	63.9	44.7	22.8	43.9	54.6
AFPTN	ResNet-50		46.5	66.1	49.1	25.6	50.2	64.2
	ResNet-101		48.3	68.0	50.9	28.3	52.3	65.5
	ResNeXt-101	✓	52.3	72.1	55.1	30.7	55.1	67.9

TTA test-time augmentation, which includes multi-scale testing, horizontal flipping, *etc*. *mstrain* multi-scale training

proposed attention feature pyramid encoder. We further visualize our failure mode in Fig. 5, mainly resulting from confusion with similar objects and low-quality images.

4.6 Running Time

Each iteration of AFPN with ResNet50 takes 855 ms on a single machine with 8 V100 cards. Thus, the total training times are about 13 days for MS COCO, respectively. During testing, each image only uses 65 ms, while the original DETR requires 43 ms per image.

4.7 Comparison with the State of the Arts

We use ResNet-50, ResNet-101 and ResNeXt-101-32x4d as the backbones for AFPN. Table 3 lists the comparison with the state-of-the-art detectors on MS COCO. The bounding box detection results for MS COCO are shown in Table 3. The results are divided into 3 groups.

The first group shows one-stage detectors. The second group shows multi-stage detectors. The third group is our results. The results can be also categorized as simple test results and TTA results, where TTA is short for test-time augmentation. The third column shows whether TTA is used. Note that different methods use different TTA strategies. For example, CBNet uses a strong TTA strategy, which can improve their box AP from 50.7 to 53.3%. Our TTA strategy brings large 4.0% AP improvement when using ResNeXt-101-32x4d as the backbone. The simple test settings can also vary significantly among different detectors. Larger input sizes tend to bring improvements. The state-of-the-art detectors are well-established and highly-optimized with sophisticated multi-stage training procedures on the challenging COCO object detection dataset. This may place AFPN at a disadvantage. Even so, AFPN is competitive.

5 Conclusion

In this paper, we have proposed a novel end-to-end Attention Feature Pyramid Network (AFPN) framework to learn detectors with hyper feature maps via transformer encoder-decoder fashion. The extensive experiments demonstrate that AFPN outperforms its baseline , which is effective on the challenging COCO dataset. For the future work, we note that there are not enough samples to train an object detectors, resulting in the few shot problem. We plan to extend our method to solve the Small-Sample-Size problem.

Acknowledgements This research is supported by the National Natural Science Foundation of China (No. 62072386, No.61866019); Yunnan provincial major science and technology special plan projects (No. 202103AA080015,No.202002AD080001); The Program For Applied & Basic Research of Yunnan Province in Key Areas (No. 2019FA023), the Candidates of the Young and Middle Aged Academic and Technical Leaders of Yunnan Province (No. 2019HB006).

References

1. Ciaparrone G, Sánchez F. L, Tabik S, Troiano L, Tagliaferri R, Herrera F (2020) Deep learning in video multi-object tracking: a survey. Neurocomputing 381:61-88
2. Lin X, Shen Y, Cai L, Ji R (2016) The distributed system for inverted multi-index visual retrieval. Neurocomputing 215:241–249
3. Yu J, Tao D, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
4. Yu J, Rui Y, Tao D (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032
5. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In: ICCV
6. Zou Z., Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: a survey. arXiv arXiv:1905.05055
7. Uijlings JRR, van de Sande KEA, Gevers T, Smeulders AWM (2013)Selective search for object recognition, IJCV
8. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv. Neural Inf Process Syst 28:91–99
9. Redmon J, Divvala S, Girshick R, Farhadi A (2016) you only look once: unified, real-time object detection. In: CVPR
10. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: ECCV
11. Girshick R (2015) Fast R-CNN. In: ICCV
12. Singh B, Davis LS (2018) An analysis of scale invariance in object detection - SNIP. In: CVPR
13. Singh B, Najibi M, Davis LS (2018) SNIPER: efficient multi-scale training. In: NeurIPS
14. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: ECCV

15. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: CVPR
16. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR
17. He K, Zhang X, Ren S, Sun J (2014) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: ECCV
18. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. In: NeurIPS
19. Shrivastava A, Gupta A, Ross G (2016) Training region-based object detectors with online hard example mining. In: CVPR
20. Cai Z, Vasconcelos N (2018) Cascade R-CNN: delving into high quality object detection. In: CVPR
21. Hu H, Gu J, Zhang Z, Dai J, Wei Y (2018) Relation networks for object detection. In: CVPR
22. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: ICCV
23. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: a metric and a loss for bounding box regression. In: CVPR
24. Wang J, Chen K, Yang S, Loy CC, Lin D (2019) Region proposal by guided anchoring. In: CVPR
25. Zhang S, Chi C, Yao Y, Lei Z, Li SZ (2020) Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: CVPR
26. Tychsen-Smith L, Petersson L (2018) Improving object localization with fitness NMS and bounded IoU loss. In: CVPR
27. Shen Y, Ji R, Chen Z, Hong X, Zheng F, Liu J, Xu M, Tian Q (2020) Noise-aware fully webly supervised object detection. In: CVPR
28. Shen Y, Ji R, Yang K, Deng C, Wang C (2019) Category-aware spatial constraint for weakly supervised detection. IEEE Trans Image Process 29:843–858
29. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K (2015) Spatial transformer networks. In: NeurIPS
30. Shen Y, Ji R, Wang Y, Wu Y, Cao L (2019) Cyclic guidance for weakly supervised joint detection and segmentation. In: CVPR
31. Shen Y, Ji R, Zhang S, Zuo W, Wang Y (2018) Generative adversarial learning towards fast weakly supervised detection. In: CVPR
32. Tian Z, Shen C, Chen H, He T (2019) FCOS: fully convolutional one-stage object detection. In: ICCV
33. Fu CY, Liu W, Ranga A, Tyagi A, Berg AC (2017) DSSD: deconvolutional single shot detector. arXiv arXiv:1701.06659
34. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: ICCV
35. Zhang S, Wen L, Bian X, Lei Z, Li SZ (2018) single-shot refinement neural network for object detection. In: CVPR
36. Li Z, Peng C, Yu G, Zhang X, Deng Y, Sun J (2017) Light-Head R-CNN: in defense of two-stage object detector. arXiv arXiv:1711.07264
37. Tychsen-Smith L, Petersson L (2017) DeNet: scalable real-time object detection with directed sparse sampling. In: ICCV
38. Singh B, Li H, Sharma A, Davis LS (2018) R-FCN-3000 at 30fps: decoupling detection and classification. In: CVPR
39. Wang R. J, Li X, Ao S, Ling CX (2018) Pelee: a real-time object detection system on mobile devices. In: NeurIPS
40. Qin Z, Li Z, Zhang Z, Bao Y, Yu G, Peng Y, Sun J (2019) ThunderNet: towards real-time generic object detection. In: ICCV
41. Ronneberger O, Fischer P, Brox T (2015) U-Net: convolutional networks for biomedical image segmentation. In: MICCAI
42. Jeong J, Park H, Kwak N (2017) Enhancement of SSD by concatenating feature maps for object detection. In: BMVC
43. Ren J, Chen X, Liu J, Sun W, Pang J, Yan Q, Tai Y.W, Xu L (2017) Accurate single stage detector using recurrent rolling convolution. In: CVPR
44. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: CVPR
45. Bell S, Zitnick CL, Bala K, Girshick R (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. In: CVPR
46. Kong T, Yao A, Chen Y, Sun F (2016) HyperNet: towards accurate region proposal generation and joint object detection. In: CVPR
47. Cai Z, Fan Q, Feris R. S, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. In: ECCV
48. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: ECCV

49. Cao J, Pang Y, Li X (2019) Triply supervised decoder networks for joint detection and segmentation. In: CVPR
50. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N (2021) An image is worth 16x16 words: transformers for image recognition at scale. In: ICLR
51. Chen H, Wang Y, Guo T, Xu C, Deng Y, Liu Z, Ma S, Xu C, Xu C, Gao W (2021) Pre-trained image processing transformer. In: CVPR
52. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NeurIPS
53. Lin T.-Y, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2014) Microsoft COCO: common objects in context. In: ECCV
54. Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: ICLR
55. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: AISTATS
56. Redmon J, Farhadi A, Ap C (2018) YOLOv3 : an incremental improvement. arXiv arXiv:1804.02767
57. Law H, Deng J (2018) CornerNet: detecting objects as paired keypoints. In: ECCV
58. Zhou X, Zhuo J, Krähenbühl P (2019) Bottom-up object detection by grouping extreme and center points. In: CVPR
59. Zhu C, He Y, Savvides M (2019) Feature selective anchor-free module for single-shot object detection. In: CVPR
60. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q (2019) CenterNet: keypoint triplets for object detection. In: ICCV
61. Ghiasi G, Lin TY, Le QV (2019) NAS-FPN: learning scalable feature pyramid architecture for object detection. In: CVPR
62. Du X, Lin T-Y, Jin P, Ghiasi G, Tan M, Cui Y, Le QV, Song X: SpineNet: Learning Scale-Permuted Backbone for Recognition and Localization. CVPR 2020: 11589–11598
63. Pang J, Chen K, Shi J, Feng H, Ouyang W, Lin D (2019) Libra R-CNN: towards balanced learning for object detection. In: CVPR
64. Zhu X, Hu H, Lin S, Dai J (2019) Deformable ConvNets v2: more deformable, better results. In: CVPR
65. Li Y, Chen Y, Wang N, Zhang Z (2019) Scale-aware trident networks for object detection. In: ICCV
66. Song G, Liu Y, Wang X (2020) Revisiting the sibling head in object detector. In: CVPR
67. Peng C, Xiao T, Li Z, Jiang Y, Zhang X, Jia K, Yu G, Sun J (2018) MegDet: a large mini-batch object detector. In: CVPR
68. Liu Y, Wang Y, Wang S, Liang T, Zhao Q, Tang Z, Ling H (2019) CBNet: a novel composite backbone network architecture for object detection. In: AAAI
69. Chen B, Medini T, Farwell J, Gobriel S, Tai C, Shrivastava A (2019) Slide: in defense of smart algorithms over hardware acceleration for large-scale deep learning systems