CF-DETR: Coarse-to-Fine Transformers for End-to-End Object Detection

Xipeng Cao,[1*] **Peng Yuan,** [2†] **Bailan Feng,** [2] **Kun Niu** [1†]

[1] Beijing University of Posts and Telecommunications
[2] Huawei Noah's Ark Lab
{xpcao,niukun}@bupt.edu.cn, {yuanpeng126,fengbailan}@huawei.com

Abstract

The recently proposed DEtection TRansformer (DETR) achieves promising performance for end-to-end object detection. However, it has relatively lower detection performance on small objects and suffers from slow convergence. This paper observed that DETR performs surprisingly well even on small objects when measuring Average Precision (AP) at decreased Intersection-over-Union (IoU) thresholds. Motivated by this observation, we propose a simple way to improve DETR by refining the coarse features and predicted locations. Specifically, we propose a novel Coarse-to-Fine (CF) decoder layer constituted of a coarse layer and a carefully designed fine layer. Within each CF decoder layer, the extracted local information (region of interest feature) is introduced into the flow of global context information from the coarse layer to refine and enrich the object query features via the fine layer. In the fine layer, the multi-scale information can be fully explored and exploited via the Adaptive Scale Fusion(ASF) module and Local Cross-Attention (LCA) module. The multi-scale information can also be enhanced by another proposed Transformer Enhanced FPN (TEF) module to further improve the performance. With our proposed framework (named CF-DETR), the localization accuracy of objects (especially for small objects) can be largely improved. As a byproduct, the slow convergence issue of DETR can also be addressed. The effectiveness of CF-DETR is validated via extensive experiments on the coco benchmark. CF-DETR achieves state-of-the-art performance among end-to-end detectors, e.g., achieving 47.8 AP using ResNet-50 with 36 epochs in the standard 3x training schedule.

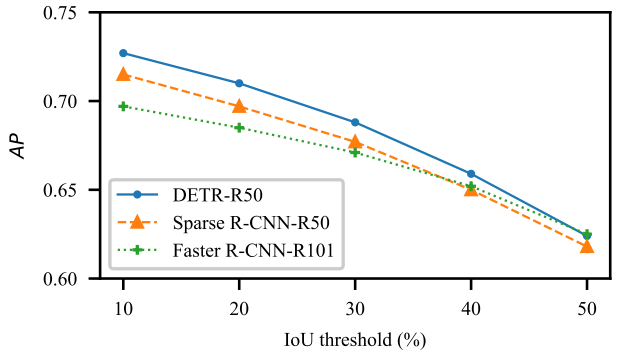
Introduction

Object detection which involves classification and localization subtasks is a fundamental problem in the field of Computer Vision (Zou et al. 2019; Zaidi et al. 2021). The modern object detectors (Liu et al. 2016; Redmon et al. 2016; Lin et al. 2020; Ren et al. 2017; He et al. 2017; Zhang et al. 2020) rely on post-processing (e.g., "non-maximum suppression" or NMS) to get robust detection results. Recently, DEtection TRansformer (DETR) (Carion et al. 2020) has been proposed as a fully end-to-end object detector, which does not rely on NMS. It utilizes object queries that contain properties (feature, shape, location, etc.), to query from a global context through a cross-attention mechanism. Although DETR achieves promising performance for end-to-end object detection, it is thought to have relatively poor detection performance on small objects and suffers from slow convergence.

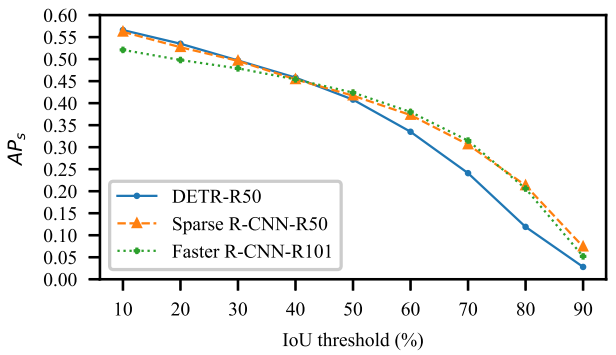
However, when observing the COCO-style metric Average Precision (AP) at different IOU thresholds, we get more insights into DETR's behavior. As illustrated in Figure 1, we calculated the AP results on COCO validation set at various IoU thresholds for three methods: DETR-R50, Sparse R-CNN-R50, and Faster R-CNN-R101. Note that the chosen three methods have similar performances measured with conventional $AP_{50:95}$. Figure 1(a) shows that DETR performs much better than other methods when measuring AP with low IoU thresholds. Figure 1(b) further shows AP results on small objects at different IoU thresholds. While DETR indeed performs poorly at high IoU thresholds (e.g. from 0.5 to 0.9), it performs surprisingly well (even better than Sparse R-CNN-R50 with FPN) when measuring AP with low IoU thresholds (e.g. from 0.1 to 0.4). We also summarize the AP scores at different ranges in Table 1, which clearly shows the superiority of DETR when measuring AP in the low IOU thresholds range. This phenomenon implies the strong perception ability of DETR even for small objects, and the reason why DETR is poorer on small objects is that the bounding box location is not accurate enough compared with Region of Interests (RoI) feature based methods (like Faster R-CNN and Sparse R-CNN). And that the key to improving DETR is simply to refine the coarse predicted locations by introducing local information.

Several methods have been proposed to explore local information in DETR architecture (Zhu et al. 2020; Sun et al. 2020; Gao et al. 2021). Deformable DETR (Zhu et al. 2020) leverages multi-scale deformable encoder and sparse sampling for local information rather than global information. Instead of using the cross-attention module, TSP (Sun et al. 2020) combines RCNN- or FCOS-based (Tian et al. 2019) methods with the Transformer encoder to focus on local information. Different from the above methods, this paper proposes a new way to fully utilize the global context information and local information in a coarse-to-fine manner. Al-

[*]This work is done when Xipeng Cao was an intern in Huawei Noah's Ark Lab.
[†]Corresponding author.



(a) AP results at IoU threshold of 0.1 to 0.5.



(b) AP results on small objects at IoU threshold of 0.1 to 0.9.

Figure 1: AP results on COCO validation set at various IoU thresholds: DETR-R50 vs. Sparse R-CNN-R50 vs. Faster R-CNN-R101. (a) shows AP results at IoU thresholds from 0.1 to 0.5. DETR performs much better than other methods under this setting. Note that all the compared methods have similar performance measured with conventional $AP_{50:95}$. (b) shows AP results on small objects at IoU thresholds from 0.1 to 0.9. While DETR indeed performs poorly at the IoU threshold from 0.5 to 0.9, it performs well when measuring AP with low IoU thresholds from 0.1 to 0.4.

Method	AP	AP_s	AP'	AP'_s
Faster R-CNN-R101	**42.5**	24.2	66.7	47.5
DETR-R50	42.0	20.5	**68.3**	**49.4**
Sparse R-CNN-R50	42.8	**26.7**	67.2	49.1

Table 1: Average Precision at different ranges of IoU thresholds for all objects and small objects. AP and AP' denotes $AP_{50:95}$ and $AP_{10:50}$ for all objects separately. AP_s and AP'_s denotes $AP_{50:95}$ and $AP_{10:50}$ for small objects separately.

though this approach is not proposed directly to solve the low convergence problem, we believe that it is in line with mitigating this issue. As the predicted boxes are refined to be more accurate, the label-assignment matching process will be more stable, therefore the training process will be more efficient (Sun et al. 2020).

In this paper, a Coarse-to-Fine DEtection TRansformer (CF-DETR) is proposed (see Figure 2), which retains the non-local encoder-decoder architecture of DETR to inherit its strong perception ability. In CF-DETR, a coarse-to-fine (CF) decoder layer constituted of a coarse layer and a fine layer, is designed to improve the localization accuracies. With the CF structure, the local multi-scale RoI information can be extracted and introduced into the flow of global attention information from the coarse layer to gradually enrich the object query features via the fine layer. In the fine layer, we propose a novel **Adaptive Scale Fusion (ASF)** module, which leverages object query features to adaptively fuse RoI features from different scales. The fused RoI features are further feed into a novel **Local Cross-Attention (LCA)** module to refine and enrich object query features via the cross-attention interactions. Compared with the vanilla cross-attention, the proposed attention module is more conducive to obtaining the local and spatial information of objects and its convergence is faster. In addition, the original multi-scale features can be enhanced by a novel **Transformer Enhanced FPN (TEF)** module, which transfers the high-level non-local information extracted from Transformer Encoder to the low-level features in an FPN manner, bringing further performance improvement of CF-DETR.

The main contributions of this paper are as follows:

- A new end-to-end object detection transformer framework named CF-DETR is proposed. In CF-DETR, a novel CF decoder layer is proposed to refine and enrich the features in a coarse-to-fine manner by fusing local and global information.
- In the fine layer, an ASF module and an LCA module are proposed to fully explore and exploit the multi-scale RoI information. In addition, a TEF module is proposed to enhance the original multi-scale information, further improving the performance of CF-DETR.
- The effectiveness of CF-DETR is demonstrated by the experimental results on the challenging COCO dataset (Lin et al. 2014). CF-DETR achieves state-of-the-art performance among end-to-end detectors, e.g., achieving 47.8 AP using ResNet-50 with 36 epochs in the standard 3x training schedule.

Related Work

One-stage and two-stage object detectors

Previous deep learning-based object detectors can be divided into two categories: one-stage and two-stage detectors. Typically, one-stage detectors such as SSD (Liu et al. 2016), YOLO (Redmon et al. 2016), and RetinaNet (Lin et al. 2020), directly conduct object classification and location on pixels of the output feature map. While two-stage detectors (Ren et al. 2017; He et al. 2017) first generate RoI based on sliding-window locations. And then they leverage the RoI align layer to extract fine-grained features and refine proposals.

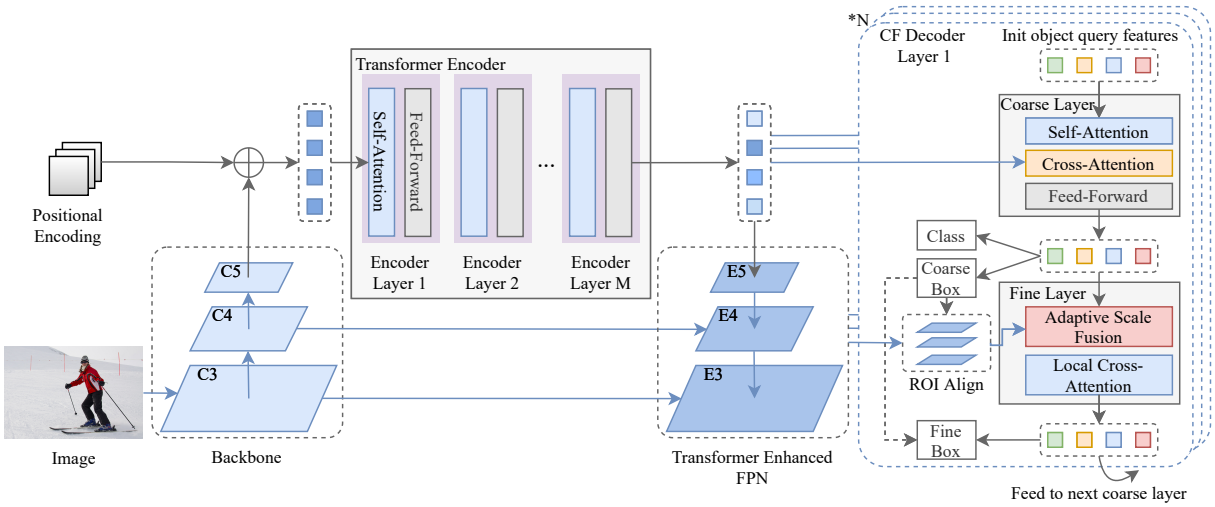


Figure 2: The overview of CF-DETR. CF-DETR follows the main encoder-decoder architecture of DETR, with a novel TEF module and novel CF decoder layers. The features from Transformer Encoder and TEF module are taken as inputs by CF decoder layers. Each CF decoder layer contains a coarse layer and a fine layer. The coarse layer follows the traditional Transformer decoder layer structure. The fine layer leverages multi-scale RoI features to refine coarse boxes from the coarse layer via the ASF and LCA modules. The object query features are passed through N cascaded CF decoder layers.

However, all these methods require hand-crafted principles (e.g., intersection-over-union (IoU) threshold) when assigning predictions to ground-truth object boxes. Moreover, the leverage of NMS post-processing (Bodla et al. 2017) to remove redundant boxes is also necessary for the inference phase.

End-to-end detectors

Recently, DEtection TRansformer (DETR) (Carion et al. 2020) has been proposed as end-to-end object detection, which utilizes Hungarian matching for label assignment. Although it achieves comparable performance with Faster RCNN, it has relatively lower detection performance on small objects and suffers from slow convergence.

To accelerate the convergence speed of DETR, Deformable DETR (Zhu et al. 2020) proposes a deformable encoder, which extracts multi-scale features naturally via learnable sparse sampling. Based on Deformable DETR, Efficient DETR (Yao et al. 2021) proposes that a great initialization of object queries could help the model converge. With a dense-to-sparse structure, Efficient DETR builds a simple yet efficient end-to-end detector with one decoder layer. TSP (Sun et al. 2020) points out that the cross-attention mechanism is the main reason for the slow convergence of DETR, such that they propose to combine RCNN- or FCOS-based (Tian et al. 2019) methods with Transformer encoders . While Deformable DETR and TSP explored local information, SMCA (Gao et al. 2021) explored global information with a self-attention and co-attention mechanism to accelerate convergence speed. On the other hand, UP-DETR (Dai et al. 2020) proposes a novel self-supervised DETR. It enhances convergence speed and performance by pre-training the Transformer encoder in DETR.

Recently, Sparse R-CNN (Sun et al. 2021) proposes a fully sparse structure with an end-to-end set prediction loss. It utilizes a dynamic interaction module to extract fine object features from local RoI features.

Different from the above methods, this paper proposes a coarse-to-fine manner to fully utilize both the global context information and local RoI information. It efficiently improves the model localization capability and only requires standard 3x training strategies to converge.

Method

The central idea of the CF-DETR framework is to refine coarse bounding boxes. With this framework, both the global context information and local RoI information can be utilized efficiently, and the multi-scale information can be enhanced and fully explored. Therefore the predicted boxes based on the refined and enriched object query features will be more accurate.

Overview

Figure 2 shows the pipeline of CF-DETR. It follows the main encoder-decoder architecture of DETR. Different from DETR, CF-DETR has a novel TEF module and novel CF decoder layers. The features from Transformer Encoder and TEF module are taken as inputs by CF Decoder layers to complete the following detection tasks. Each CF decoder layer contains a coarse layer and a fine layer. The coarse layer extracts object-related features from global context semantics. And the fine layer leverages multi-scale RoI features to refine coarse boxes from the coarse layer via the ASF and LCA modules. The object query features are passed through the cascaded CF decoder layers and are optimized together with network parameters.

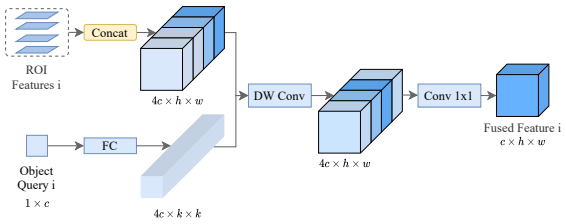


Figure 3: illustrates the details of the Adaptive Scale Fusion module.

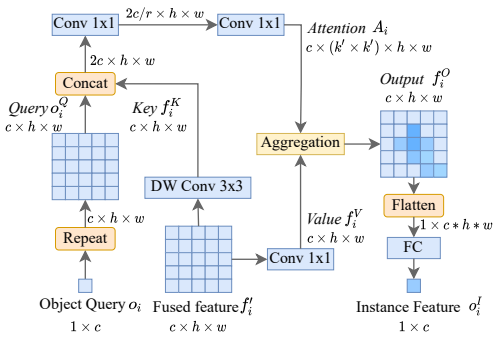


Figure 4: illustrates the details of the Local Cross-Attention module.

Modules

TEF Module. With the Transformer Encoder in DETR or CF-DETR, the non-local foreground features $E5$ can be extracted from the backbone features $C5$. We expect this non-local foreground information could be transmitted to low-level features to help improve the perception of objects. Inspired by FPN (Lin et al. 2017a), we propose a TEF module that works the same way as FPN, except that the features C5 is replaced by the output features E5 from Transformer Encoder. Specifically, we first add the upsampled $E5$ to $C4$, then output the fused feature map $E4$ after 3×3 convolution. A new set of feature maps $\{Ei\}_{i=1}^{L}$ can be obtained by repeating the above operation between adjacent feature maps, where L is a hyper-parameter. With the TEF module, the multi-scale information can be enhanced, also the gap between the features from the coarse layer and RoIs might be alleviated, which may benefit the cross-attention operations in the fine layer.

Coarse-to-fine Decoder Layer. As illustrated in Figure 2, the CF decoder layers are cascaded and they take as inputs a set of learnable object query features and the features from the TEF module to detect objects. Each CF decoder layer contains a coarse layer and a fine layer:

(1) ***Coarse Layer***: The coarse layer follows the traditional Transformer decoder layer structure, which could be also replaced with other variants (Gao et al. 2021; Meng et al. 2021) of DETR, extracting object-related features from global context semantics. The self-attention module first embeds the relationships between N object queries as $O \in \mathbb{R}^{N \times c}$. Given a flattened feature map $x \in \mathbb{R}^{HW \times c}$ from Transformer encoder, each object $o_i \in \mathbb{R}^{1 \times c}$ focuses on different regions of feature map x to sense an potential object via the cross-attention module. Note that learnable object query features are directly fed as queries into the cross-attention module rather than treated as positional encoding in DETR. FFN layers and other layer norms are added into the pipeline similar to the Transformer setting. The bounding box predictions are computed by a 3-layer multilayer perception (MLP) with ReLU activation functions. Classification is performed by a single linear layer.

(2) ***Fine Layer***: The goal of the fine layer is to further refine the coarse bounding boxes. It takes as inputs the object query features which contain global context information output from the previous coarse layer and the local RoI features extracted from the TEF module. Object query features containing global information are sufficient for the classification task (Wang et al. 2020). However, more precise local information (e.g. the shape and boundaries of objects) and multi-scale information are necessary for accurate location predictions, especially for small objects. Thus the fine layer further explores and exploits the multi-scale local RoI features via the proposed ASF and LCA modules.

ASF Module. Given one object query $o_i \in \mathbb{R}^{1 \times c}$ and related coarse box b_i from the coarse layer, the fine layer utilizes the RoI align (He et al. 2017) operation to extract corresponding multi-scale features $\{f_i^l \in \mathbb{R}^{c \times h \times w}\}_{l=1}^{L}$ from the TEF module, where L is the number of levels. The conventional method uses the heuristic method to select a specific layer of RoI feature based on the size of b_i. This method does not fully explore high-level semantic information. Other works (Liu et al. 2018; Guo et al. 2020) have attempted to aggregate RoI features to improve performance, but these methods do not consider the corresponding instance feature o_i. Here, we propose an ASF module to fuse multi-scale features adaptively according to specific object query features (see Figure 3). Specifically, all RoI features in different scales are concatenated in channel dimensions $f_i \in \mathbb{R}^{Lc \times h \times w}$. The convolution weights with spatial size $k \times k$ are generated according to the object query feature via a fully connected layer. Then the ASF leverages depth-wise convolution to activate informative channels for the object. Finally, a 1×1 convolution is applied for reducing dimension from Lc to c.

LCA Module. The fused multi-scale RoI information from ASF is exploited by object queries in the LCA module, which basically implements a Local Cross-Attention (LCA) between the object query features and the fused RoI features. As spatial information is very important for precisely locating an object, different from the non-local multi-attention mechanism, LCA enables object queries focusing on local information when interacting with the fused RoI features. Specifically, for a given pair of object query feature $o_i \in \mathbb{R}^{1 \times c}$ and fused feature $f_i' \in \mathbb{R}^{c \times h \times w}$, L CA first employs a depth-wise 3×3 convolution over f_i' for extracting contextual information from local neighbor points, producing the key: $f_i^K = DWConv_{3\times3}(f_i')$, $f_i^K \in \mathbb{R}^{c \times h \times w}$. The query is defined as the expand of object query feature: $o_i^Q \in \mathbb{R}^{c \times h \times w}$, with the same shape as f_i^K. Inspired by CoT-

Method	Feature	Epochs	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
Faster R-CNN-R50 (Wu et al. 2019)	FPN	36	40.2	61.0	43.8	24.2	43.5	52.0	26
Cascade R-CNN-R50 (Wu et al. 2019)	FPN	36	44.3	62.2	48.0	26.6	47.7	57.7	19
DETR-R50 (Carion et al. 2020)	Encoder	500	42.0	62.4	44.2	20.5	45.8	61.1	28
DETR-DC5-R50 (Carion et al. 2020)	Encoder	500	43.3	63.1	45.9	22.5	47.3	61.1	12
Deform DETR*-R50 (Zhu et al. 2020)	DeformEncoder	50	43.8	62.6	47.7	26.4	47.1	58.0	19
Deform DETR*++-R50 (Zhu et al. 2020)	DeformEncoder	50	46.2	65.2	50.0	28.8	49.2	61.9	19
Sparse R-CNN-R50 (Sun et al. 2021)	FPN	36	42.8	61.2	45.7	26.7	44.6	57.7	23
Sparse R-CNN*-R50 (Sun et al. 2021)	FPN	36	45.0	63.4	48.2	26.9	47.2	59.5	22
TSP-RCNN-R50 (Sun et al. 2020)	FPN	36	43.8	63.3	48.3	28.6	46.9	55.7	11
SMCA*-R50 (Gao et al. 2021)	Encoder	108	45.6	65.6	49.1	25.9	49.3	62.6	10
CF-DETR-R50	TEF	36	46.5	65.2	50.5	28.4	49.3	61.8	18
CF-DETR*-R50	TEF	36	**47.8**	**66.5**	**52.4**	**31.2**	**50.6**	**62.8**	16
Faster R-CNN-R101 (Wu et al. 2019)	FPN	36	42.0	62.5	45.9	25.2	45.6	54.6	20
DETR-R101 (Carion et al. 2020)	Encoder	500	43.5	63.8	46.4	21.9	48.0	61.8	20
DETR-DC5-R101 (Carion et al. 2020)	Encoder	500	44.9	64.7	47.7	26.4	47.1	58.0	10
Sparse R-CNN-R101 (Sun et al. 2021)	FPN	36	44.1	62.1	47.2	26.1	46.3	59.7	19
Sparse R-CNN*-R101 (Sun et al. 2021)	FPN	36	46.4	64.6	49.5	28.3	48.3	61.6	18
TSP-R-CNN-R101 (Sun et al. 2020)	FPN	36	44.8	63.8	49.2	29.0	47.9	57.1	9
SMCA*-R101 (Gao et al. 2021)	Encoder	50	44.4	65.2	48.0	24.3	48.5	61.0	-
CF-DETR-R101	TEF	36	47.2	65.9	51.1	29.0	50.2	63.4	16
CF-DETR*-R101	TEF	36	**49.0**	**68.1**	**53.4**	**31.4**	**52.2**	**64.3**	14

Table 2: Evaluation results of related methods on COCO 2017 val set. The results of other methods are from Detectron2 (Wu et al. 2019) and their released papers. Note that "*" indicates that there are 300 object queries in training. Deform DETR*++ means Deformable DETR with the two-stage trick. A single NVIDIA Tesla V100 GPU is used to measure the inference time.

Net (Li et al. 2021), the attention map is calculated as follow:

$$A_i = (o_i^Q, f_i^K)W_1W_2 \tag{1}$$

where $W_1 \in \mathbb{R}^{2c\times 2c/r}$ and $W_2 \in \mathbb{R}^{2c/r\times(c*k'*k')}$ are parameters of 1×1 convolution kernels, and r is the dimension scaling factor. When we do attention aggregation, vectors within each k' size of value matrix are weighted together, which is shown as follows:

$$f_i^O(h',w') = \sum_{u=1}^{k'}\sum_{v=1}^{k'} A_{i,u,v,h',w'} \odot f_i^V(h',w')_{u,v} \tag{2}$$

where $f_i^V \in \mathbb{R}^{c\times h\times w}$ is projected from fused feature f_i' via 1×1 convolution. And $f_i^V(h',w')_{u,v}$ means the neighbor point (u,v) of point (h',w') on the value matrix f_i^V. Ai,u,v,h',w' is the corresponding attention vector on attention map A_i. Here $\odot$ means the Hadamard product. In the end, f_i^O is flattened to the shape $1\times(c*h*w)$ and its dimension is reduced to $1\times c$ through one FC layer. The refine and enriched object query features are then taken as inputs by a 3-layer MLP with ReLU activation functions to predict locations.

Iterative structure. The prediction of the bounding boxes is in an iterative structure. Specifically, within first CF decoder layer, the fine layer refines the bounding boxes based on the outputs (the normalized center coordinates, heights and widths of the boxes for a given image) of the coarse layer. Also, the CF decoder layers are cascaded to further improve the performance. The refined object query features from the last fine layer will be sent to the next coarse layer. The new coarse layer also refines the boxes based on the output of the previous fine layer (Cai and Vasconcelos 2018).

Loss. The proposed CF-DETR aligns set prediction loss with DETR. Note that, as the fine layer only predicts bounding boxes, the predicted class logits from the coarse layer are used when calculating the matching cost. After label assignment, the total detection loss can be written as follows:

$$\begin{aligned}\mathcal{L}_{det} = \lambda_{cls}\cdot\mathcal{L}_{cls} + \lambda_{L_1}\cdot\mathcal{L}^c_{L_1} + \lambda_{L_1}\cdot\mathcal{L}^f_{L_1} \\ \lambda_{giou}\cdot\mathcal{L}^c_{giou} + \lambda_{giou}\cdot\mathcal{L}^f_{giou},\end{aligned} \tag{3}$$

where $\mathcal{L}_{cls}$ is focal loss (Lin et al. 2017b) for coarse layers classifications. $\mathcal{L}^c_{L_1}$ and $\mathcal{L}^f_{L_1}$ are L1 losses of predicted bounding boxes for coarse and fine layers seperately. $\mathcal{L}^c_{giou}$ and $\mathcal{L}^f_{giou}$ are generalized IoU losses for coarse and fine layers seperately . λ_{cls}, λ_{L_1} and λ_{giou} are trade-off hyperprameters for each loss.

Experiments

Dataset and Evaluation Metrics

MS COCO (Lin et al. 2014) instance detection dataset is utilized to evaluate detectors. Where all models are trained on the COCO train2017 set with 118k images and evaluated on the val2017 set with 5k images. The performances on COCO 2017 test-dev set are also reported. Following the common practice, AP on the coco val2017 set is used as the main metric. To verify whether CF-DETR mitigates the defects of the DETR, we also focus on APs with small objects and APm for medium objects. The convergence speed is also concerned.

Method	Backbone	TTA	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
TSP-R-CNN (Sun et al. 2020)	ResNet-101+DCN		47.4	66.7	51.9	29.0	49.7	59.1
Sparse R-CNN (Sun et al. 2021)	ResNeXt-101+DCN	✓	51.5	71.1	57.1	34.2	53.4	64.1
Deformable DETR (Zhu et al. 2020)	ResNeXt-101+DCN	✓	52.3	71.9	58.1	34.4	54.4	65.6
CF-DETR	ResNet-50		48.1	67.2	52.5	29.5	50.0	61.3
CF-DETR	ResNet-101		49.3	68.5	53.8	29.9	51.6	63.1
CF-DETR	ResNeXt-101		49.8	69.0	54.4	31.0	52.2	63.0
CF-DETR	ResNeXt-101+DCN		50.7	69.9	55.4	30.7	53.2	65.4
CF-DETR	ResNeXt-101+DCN	✓	53.0	72.6	58.8	35.1	54.9	65.9

Table 3: Comparison of CF-DETR with state-of-the-art end-to-end detectors on COCO 2017 test-dev set. Note that, Sparse R-CNN, Deformable DETR, and CF-DETR are trained with 300 object queries. "TTA" indicates test-time augmentations.

Implementation Details

Transformer Enhanced FPN. We utilize ResNet-50 and ResNet-101 (He et al. 2016) as backbones, which are pre-trained on ImageNet (Deng et al. 2009). Where feature maps $\{C_2, C_3, C_4, C_5\}$ from ResNet are feed to TEF module to extracted pyramid-like feature maps $\{E_2, E_3, E_4, E_5\}$. The channel size of feature maps is 256. The dimension of the learnable object query feature is also 256. The Transformer encoder (a 6-layer encoder of width 256 with 8 attention heads, 2048 FFN) is the same with DETR.

CF decoder layers. The number of CF decoder layers is set to 6 by default. The settings of coarse layers are the same as the Transformer decoder in DETR. In the fine layer, The shape of RoI feature maps is $256 \times 7 \times 7$. The spatial size k in the ASF module is set to 3. And the dimension scaling factor r in the LCA is set to 4. The default number of object queries is 100.

Training details. The AdamW (Loshchilov and Hutter 2019) optimizer with weight decay 1e-4 is adopted in the training process. CF-DETR is trained on 8 NVIDIA Tesla V100 GPUs, and the batch size is 16 in total. We follow the default $3\times$ training schedule of Detectron2 (Wu et al. 2019) and the initial learning rate is set to 1×10^{-4}. Data augmentations and trade-off hyperparameters in detection loss are the same with DETR.

Main Result

We compared CF-DETR with well-established detectors, such as Faster R-CNN (Ren et al. 2017), Cascade R-CNN (Cai and Vasconcelos 2018), as well as the most related end-to-end detectors: DETR (Carion et al. 2020), Deformable DETR (Zhu et al. 2020), Sparse R-CNN (Sun et al. 2021), TSP-RCNN (Sun et al. 2020), SMCA (Gao et al. 2021).

Table 2 shows that our proposed CF-DETR outperforms all the other competitors. For instance, as an end-to-end detector, CF-DETR*-R50 performs much better than the two-stage detector Cascade R-CNN-R50 measuring with AP (47.8 AP vs. 44.3 AP). Compared with Sparse R-CNN, a simple and efficient end-to-end method, our proposed method exhibits much higher AP scores (47.8 AP vs 45.0 AP with R50; 49.0 AP vs. 46.4 AP with R101). Compared with other SOTA DETR-like detectors: CF-DETR*-R50 with 36 epochs performs even better than Deformable DETR*++-R50 with 50 epochs (47.8 AP vs. 46.2 AP).

Coarse	Fine	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
✓		30.0	54.2	29.2	10.6	32.0	49.4
	✓	39.9	56.9	42.9	24.0	41.6	54.7
✓	✓	**46.5**	**65.2**	**50.5**	**28.4**	**49.3**	**61.8**

Table 4: Ablation studies on the coarse and fine layers in CF decoder layers.

TEF	ASF	LCA	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
			42.3	61.7	45.5	26.1	44.8	56.9
✓			43.8	63.5	47.4	26.5	46.6	58.8
	✓		44.2	63.1	48.2	27.8	47.2	59.1
		✓	43.4	62.6	47.0	26.8	46.3	57.6
✓	✓		44.8	63.6	48.8	27.3	47.7	59.9
✓		✓	44.7	64.1	48.7	28.1	47.7	59.5
	✓	✓	44.6	63.4	48.4	28.0	47.2	59.8
✓	✓	✓	**46.5**	**65.2**	**50.5**	**28.4**	**49.3**	**61.8**

Table 5: Ablation studies on the contributions of each module (TEF, ASF, and LCA) in the proposed CF-DETR.

CF-DETR also shows a significant advantage on small object detections. For instance, CF-DETR*-R50 improves the SOTA AP_s from 28.8 (Deformable DETR) to 31.2. This illustrates the benefits of merging global context information with local information in CF-DETR. Note that, the advanced operations (e.g. deformable convolutions) in DETR variants (Meng et al. 2021; Yao et al. 2021) can also be adopted in this framework by replacing coarse layers to further improve the performance. We leave it in future works.

Table 3 compares the proposed method with other SOTA end-to-end methods on COCO 2017 test-dev set. With ResNet-101 and ResNeXt-101 (Xie et al. 2017), the proposed method achieves 49.3 AP and 49.8 AP, respectively. By using ResNeXt-101 with DCN (Zhu et al. 2019), the performance further improves to 50.7 AP, and 53.0 AP with test-time augmentations.

Ablation Studies

Analysis of Coarse-to-Fine Structure. In this section, We further analyze the effects of the key components (coarse layer and fine layer) in the CF decoder layer. To this end, the

L	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	FPS
1	35.8	52.1	38.4	19.0	37.9	47.9	26
2	43.4	61.3	46.9	25.4	45.9	59.0	24
4	46.1	64.7	50.1	28.5	48.8	61.2	21
6	46.5	65.2	50.5	28.4	49.3	61.8	18
12	45.7	64.6	49.8	28.2	48.3	60.7	13

Table 6: The AP scores and FPS of CF-DETR (with R50 backone and 100 object queries) for different numbers of CF decoder layers.

performances of CF-DETR with different implementations of CF decoder layer (with coarse layer only, with fine layer only, and with both of them) are compared (see Table 4).

Specifically, the CF decoder layers implemented with only coarse layer almost degenerate to the Transformer decoder layer in the original DETR. The only difference lies in that the predicted bounding boxes at each layer are independent in the original DETR, while we instead utilize an iterative structure in CF-DETR, where the predictions of the current layer are based on the predicted bounding boxes of the previous layer. For the CF decoder layer implemented with the fine layer only, a classification head is added, as the vanilla fine layer only predicts bounding boxes in the default design. For fair comparisons, both of them are compared with default CF decoder layers (with both coarse and fine layers) under the same feature extractor setting.

As shown in Table 4, measured with AP, the fine layer performs better (39.9 AP vs. 30.0 AP), which indicates that the fine layer is more promising at accurate localization. Combing both coarse and fine layers, the CF decoder layer achieves 16.5 AP and 5.6 AP score improvements compared with only coarse layer and only fine layer respectively (46.5 AP vs. 30.0 AP, 46.5 AP vs. 39.9 AP). This validates the effectiveness of the CF decoder layer designed with the coarse-to-fine structure.

Influences of Different Modules of CF-DETR. In this part, we further analyze the contributions of proposed modules (TEF, ASF, LCA) in CF-DETR (see Table 5). We first build a baseline model following the CF-DETR framework by replacing TEF with conventional FPN, replacing ASF with conventional heuristic layer selection method, and replacing LAC with a simple single FC layer. Then, we add the above modules one by one to see their contributions more clearly.

The baseline again demonstrates the advantage of the coarse-to-fine structure of CF-DETR, which achieved better performance than DETR-R50 (42.3 AP vs. 42.0 AP). Compared with baseline, TEF (+1.5 AP), ASF(+ 1.9 AP), and LCA (+1.1 AP) modules all bring improvements. Among them, the ASF contributes the most. Combining any two modules, lead to further improvements (+2.4 AP on average). And the complete CF-DETR (combining all the modules together) performs the best (+ 4.2 AP). Note that, the proposed modules not only improve small object detection but also improve the detection significantly for medium objects and large objects.

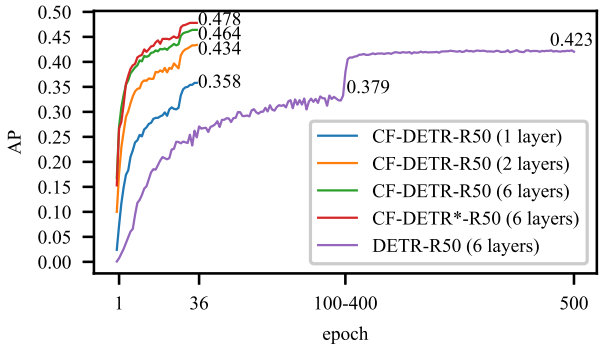


Figure 5: The convergence curves of CF-DETR (with different training settings) and DETR-R50. The CF-DETR models were trained with a standard 3x schedule. DETR-R50 was trained with 500 epochs, and the learning rate dropped after 400 epochs.

Numbers of CF decoder layers. In this section, we study the effect of the different decoder layers on the performances. As illustrated in Table 6, with only two decoder layers, CF-DETR can achieve competitive performance as DETR-DC-R50 (43.3 AP) and TSP R-CNN-R50 (43.8 AP). As the number of layers increases, the performance of CF-DETR generally improves accordingly, but the inference speed becomes slower. However, the performance decreased with 12 decoder layers. we infer that more data and iterations may be required for models with 12 decoder layers.

Analysis of convergence.

Another gain that the proposed framework brings is the fast convergence. Figure 5 compares the convergence curves of CF-DETR (with different training settings) with that of DETR. The possible reasons for faster convergence are as follows: (1) Due to the integration of global and local information via CF decoder layers, the object query features are refined and enriched. This facilitates the sparseness of the attention weight matrix in the cross-attention layer. (2) As illustrated in Figure 5, we find that the iterative structure also leads to fast convergence, as the proper cascade layers the better convergence speed we can observe.

Conclusion

This paper proposes a new end-to-end object detection transformer framework named CF-DETR. In CF-DETR, a novel CF decoder layer is proposed to refine predictions and enrich the features in a coarse-to-fine manner. To fuse local and global information efficiently in CF-DETR, an ASF module and an LCA module are proposed to fully explore and exploit the multi-scale RoI information. In addition, the multi-scale features are further enhanced by a proposed TEF module. CF-DETR achieves state-of-the-art performance among end-to-end detectors. We hope the work of this paper could inspire more insights for improving DETR-like detectors.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Grant No.61971066).

References

Bodla, N.; Singh, B.; Chellappa, R.; and Davis, L. S. 2017. Soft-NMS - Improving Object Detection with One Line of Code. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 5562–5570. IEEE Computer Society.

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 6154–6162. IEEE Computer Society.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J., eds., *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, 213–229. Springer.

Dai, Z.; Cai, B.; Lin, Y.; and Chen, J. 2020. UP-DETR: Unsupervised Pre-training for Object Detection with Transformers. *CoRR*, abs/2011.09094.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 248–255. IEEE Computer Society.

Gao, P.; Zheng, M.; Wang, X.; Dai, J.; and Li, H. 2021. Fast Convergence of DETR with Spatially Modulated Co-Attention. *CoRR*, abs/2101.07448.

Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; and Pan, C. 2020. AugFPN: Improving Multi-Scale Feature Learning for Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 12592–12601. IEEE.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2980–2988. IEEE Computer Society.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778. IEEE Computer Society.

Li, Y.; Yao, T.; Pan, Y.; and Mei, T. 2021. Contextual Transformer Networks for Visual Recognition. *CoRR*, abs/2107.12292.

Lin, T.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017a. Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 936–944. IEEE Computer Society.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017b. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2999–3007. IEEE Computer Society.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(2): 318–327.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In Fleet, D. J.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, 740–755. Springer.

Liu, S.; Qi, L.; Qin, H.; Shi, J.; and Jia, J. 2018. Path Aggregation Network for Instance Segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 8759–8768. IEEE Computer Society.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I*, volume 9905 of *Lecture Notes in Computer Science*, 21–37. Springer.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional DETR for Fast Training Convergence. *CoRR*, abs/2108.06152.

Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 779–788. IEEE Computer Society.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6): 1137–1149.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; and Luo, P. 2021. Sparse R-CNN: End-to-End Object Detection With Learnable Proposals. 14454–14463.

Sun, Z.; Cao, S.; Yang, Y.; and Kitani, K. 2020. Rethinking Transformer-based Set Prediction for Object Detection. *CoRR*, abs/2011.10881.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 9626–9635. IEEE.

Wang, H.; Wu, X.; Huang, Z.; and Xing, E. P. 2020. High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 8681–8691. IEEE.

Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; and Girshick, R. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Xie, S.; Girshick, R. B.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 5987–5995. IEEE Computer Society.

Yao, Z.; Ai, J.; Li, B.; and Zhang, C. 2021. Efficient DETR: Improving End-to-End Object Detector with Dense Prior. *CoRR*, abs/2104.01318.

Zaidi, S. S. A.; Ansari, M. S.; Aslam, A.; Kanwal, N.; Asghar, M. N.; and Lee, B. A. 2021. A Survey of Modern Deep Learning based Object Detection Models. *CoRR*, abs/2104.11892.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 9756–9765. IEEE.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable ConvNets V2: More Deformable, Better Results. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 9308–9316. Computer Vision Foundation / IEEE.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *CoRR*, abs/2010.04159.

Zou, Z.; Shi, Z.; Guo, Y.; and Ye, J. 2019. Object Detection in 20 Years: A Survey. *CoRR*, abs/1905.05055.