EF-DETR: A Lightweight Transformer-Based Object Detector With an Encoder-Free Neck

Siyi Cheng, Jingnan Song, Mingliang Zhou, Xuekai Wei, Huayan Pu, Jun Luo, and Weijia Jia, *Fellow, IEEE*

***Abstract*—Object detection plays a key role in helping to enable industrial quality control and safety monitoring. This article introduces a lightweight and efficient transformer-based object detection network called the encoder-free DEtection TRansformer (EF-DETR). This novel architecture enhances the DETR model through a redesigned network structure, leading to improved accuracy in object detection and a more lightweight network. To address the issue of suboptimal object detection accuracy, especially for small objects in the DETR model, we introduce a multiscale feature extractor and a high-efficiency feature fusion module. These components facilitate the direct extraction of fine-grained features, thereby enabling effective object detection. Departing from the use of a high-complexity encoder structure, we explore the utilization of an encoder-free neck structure to reduce the network's computational complexity. In addition, to expedite convergence, denoising training is incorporated into the decoder. This article presents extensive experiments, and the EF-DETR demonstrates strong performance on the MS COCO2017 dataset compared to other popular models.**

***Index Terms*—DEtection TRansformer (DETR), denoising training, encoder-free, lightweight, object detection.**

Manuscript received 3 June 2024; accepted 7 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62176027, in part by the Key Projects of Basic Strengthening Plan under Grant 2022-JCJQ-ZD-018-11, in part by Chongqing Talent under Grant cstc2024ycjh-bgzxm0082, in part by the Joint Equipment Pre Research and Key Fund Project of the Ministry of Education under Grant 8091B012207, in part by the Human Resources and Social Security Bureau Project of Chongqing under Grant cx2020073, in part by Guangdong Oppo Mobile Telecommunications Corporation Ltd. under Grant H20240164, and in part by the Central University Operating Expenses under Grant 2024CDTGF-044. Paper no. TII-24-2758. *(Corresponding author: Mingliang Zhou.)*

Siyi Cheng, Jingnan Song, Mingliang Zhou, and Xuekai Wei are with the School of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: csy@stu.cqu.edu.cn; sjn@stu.cqu.edu.cn; mingliangzhou@cqu.edu.cn; xuekaiwei2-c@my.cityu.edu.hk).

Huayan Pu and Jun Luo are with the State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China (e-mail: phygood_2001@shu.edu.cn; luojun@cqu.edu.cn).

Weijia Jia is with the BNU-UIC Institute of Artificial Intelligence and Future Networks, Beijing Normal University Zhuhai, Zhuhai 519087, China, and also with the Guangdong Key Lab of AI Multi-Modal Data Processing BNU-HKBU United International College Zhuhai, Zhuhai 519087, China (e-mail: jiawj@bnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2024.3431044.

Digital Object Identifier 10.1109/TII.2024.3431044

I. Introduction

OBJECT detection is an important and fundamental task of computer vision (CV) that involves recognizing objects in an image and determining their locations, as shown in Fig. 1. This task has been widely used in many industrial applications, such as defect detection [1], [2], face detection [3], automatic driving [4], surveillance systems [5], [6], and remote sensing object detection [7], [8], [9]. Despite the significant progress achieved in recent years, object detection remains a challenging problem due to the inherent appearance, shape, and pose variability across different object categories.

With the development of deep learning, classic detectors have been mainly based on convolutional neural networks (CNNs), which can be categorized into one-stage detectors and two-stage detectors. However, this changed when Carion et al. [10] proposed the DEtection TRansformer (DETR) model, which is an end-to-end object detection model that first introduced the transformer [11] and self-attention mechanisms to an object detector. The DETR treats the task of object detection as a set prediction problem, predicting a set containing all objects at once, and the model eliminates postprocessing steps, such as nonmaximum suppression (NMS) and removes prior knowledge, such as anchors. Specifically, the DETR focuses on the information of potential objects by means of learnable object queries and an attention mechanism, which enables the extraction of specific object information from global information and analyses the object information in each object query by using a feedforward neural network (FFN) to obtain the set of predicted results. Finally, the Hungarian matching algorithm is used to assign corresponding labels to the result set. This approach greatly simplifies the pipeline for object detection.

Despite its promise, the DETR still exhibits shortcomings in terms of both its small object detection performance and its training convergence speed.

Many studies have been performed to address these two shortcomings. However, these methods often come at the cost of increased computational complexity. In this article, we propose a new model called the encoder-free DETR (EF-DETR) model, which streamlines the architecture through effective structural integration and simultaneously enhances the training convergence and small-object detection capabilities of the DETR. Our approach seeks to strike a delicate balance between performance enhancement and computational simplicity, and it provides a more viable solution for practical application in object detection tasks.

Fig. 1. Inference results obtained by the EF-DETR.

The DETR model incorporates a complex structure that comprises several transformer encoders and decoders. The encoders are responsible for encoding sequence inputs, with each layer consisting of a multiheaded self-attention mechanism and an FNN. However, these operations entail a significant number of matrix calculations, making the encoders highly computationally demanding for lengthy sequences. Therefore, we explore the possibility of removing the encoders and replacing their function with other parts of the model that are much less expensive. Nevertheless, removing encoders poses a challenge in terms of extracting suitable image features for detection tasks. To address this issue, we employ a multiscale feature extractor that can extract features at different scales. In addition, to fuse the fine-grained features that are optimal for detection, we incorporate a high-efficiency feature fusion (HEFF) module after the feature extraction process.

The slower convergence rate of the DETR can be attributed to the instability of its bipartite graph matching process. To accelerate the convergence, we can use denoising training in the decoder to stabilize this matching process. Our contributions can be summarized as follows.

1) We utilize a multiscale feature extractor to adaptively weight the features at various scales according to the object scale and add an HEFF module behind it to fuse the multiscale features. By simply changing the network connections, the detection performance, including small objects, is improved with essentially no significant increase in the computational effort needed by the original model, which results in a relatively low demand for computational resources during model execution, enabling more efficient object detection in industrial environments.
2) We keep only the transformer decoder as the neck of the detector because the multiscale feature extractor with HEFF can directly extract the fine-grained features that are suitable for object detection, therefore, the encoder-free neck structure of the EF-DETR is computationally efficient, helping alleviate computing resource requirements when deploying and running models in industrial environments.
3) To address the instability of the bipartite graph matching process in the DETR decoder, we utilize denoising training. We input the GT with a noise offset into the decoder and skip the bipartite graph matching process to directly learn the noise offset.
4) The proposed approach seeks to strike a delicate balance between improving performance and simplifying computation, and we achieve SOTA performance on the MS COCO2017 dataset with a competitive number of parameters and GFLOPs. In general, we aim to improve the detection accuracy without increasing the model complexity by optimizing the network structure.

The rest of this article is organized as follows. In Section II, we present a comprehensive review of the related work. In Section III, we provide a detailed description of the proposed method, and in Section IV, we report the experimental results. Finally, Section V concludes this article.

II. Related Work

This section is divided into three parts. Section II-A provides a partial review of classical deep learning-based object detectors, and Section II-B explores the application of transformers in the field of CV. Section II-C introduces the transformer-based detector and offers a technical comparison with our proposed method.

A. CNN-Based Object Detectors

Traditional object detection methods are built on handcrafted features and shallow trainable architectures, making it easy for performance to stagnate. With the success of deep learning in the field of image classification [12], many excellent deep learning networks have emerged [13], [14], [15] and attracted widespread use in the field of object detection. Currently, deep learning-based object detectors have surpassed traditional object detectors and have become the mainstream methods for object detection.

Deep learning-based object detectors can be classified into two categories: two-stage detectors and one-stage detectors. Two-stage detectors mainly focus on selective region proposal strategies via complex architectures; however, one-stage detectors focus on all spatial region proposals for the possible detection of objects via relatively simpler architectures in one shot [16]. Two-stage object detection methods include the faster RCNN [17], mask RCNN [18], and cascade RCNN [19]. Algorithms first generate a series of region proposals, then, the ROI alignment [18] layer extracts the fine-grained features and performs classification. One-stage methods include RetinaNet [20], YOLO [21], and its variants [22], which view the object detection problem as a regression problem without generating region proposals in the given image and directly perform object classification and localization. Both two-stage and one-stage methods require complex postprocessing steps to generate their final predictions.

Despite achieving outstanding performance, these methods all use handcrafted components to make predictions relative to the initial guesses, and the final performance of these methods depends heavily on the exact setup of these initial guesses. Therefore, end-to-end optimization cannot be performed due to the above-mentioned drawbacks.

B. Vision Transformers

Since the introduction of the transformer, an increasing number of works have focused on applying transformers to CV. The development of transformers in CV can be divided into

three main stages. In the first stage, such as Bello et al. [23] incorporated attention mechanisms into CNNs, which addressed the problem that CNNs can extract only local information and cannot extract global information. In the second stage, works, such as [10] and [24] used transformers to completely replace CNNs when resolving the image domain problem.

Many studies have explored the optimization of the Vision Transformer details. Wang et al. [25] proposed a CNN-free pyramid vision transformer that significantly reduces the computational load by incorporating a pyramid structure into the transformer. As transformers mature in CV, many studies have used them instead of deep learning convolution operations. Tang et al. [26] proposed a multiscale adaptive transformer called MATR, which employs an adaptive transformer to enhance global semantic extraction for fusing multimodal medical images, and Kaselimi et al. [27] proposed a multilabel visual converter model, ForestViT, for deforestation detection, which exploits the self-attention mechanism and avoids any convolution operations involved in commonly used deep learning models.

Recently, Liu et al. [28] proposed the Swin Transformer, which employs a local attention mechanism based on hierarchical sliding windows to support the processing of longer sequences. Shift operations allow neighboring windows to interact, which greatly reduces the computational complexity of the transformers. In addition, the hierarchical structure of the Swin Transformer allows it to extract multiscale features, which is beneficial for object detection.

C. DETR-Based Detectors

Carion et al. [10] proposed the first transformer-based detector, called the DETR, which regards the detection problem as a set prediction problem to achieve end-to-end object detection. The main advantage of the DETR is that it eliminates the dependence on handcrafted modules, such as region proposal networks and NMS, which are commonly used in classic object detectors. However, the DETR has a long convergence time and poor performance in small object detection. Zhu et al. [29] proposed the deformable DETR, in which the attention module focuses only on a few key sampling points around the reference point. The deformable DETR can achieve better performance than the DETR (especially for small objects) with 10× less training time. Dai et al. [30] proposed UP-DETR, which uses unsupervised pretraining on the transformer parameters for higher precision. Meng et al. [31] proposed the conditional DETR, which uses an improved decoder cross-attention mechanism to accelerate the training convergence process, and Gao et al. [32] proposed SMCA, a plug-and-play simple co-attention model that accelerates DETR convergence and improves the precision by introducing the Gaussian-distributed weights of the objects to be detected into the co-attention mechanism, enabling the decoder to locate the object features to be detected among the global features, thus accelerating the DETR convergence. Liu et al. [33] proposed the use of a dynamic anchor box for the DETR, which directly outputs the coordinates of the box as queries for the transformer decoder and dynamically updates the box at each layer, thus accelerating the convergence procedure.

Although these works are dedicated to improving the convergence speed and detection accuracy of the DETR, they usually come at the cost of increasing the model's computational complexity and do not find a balanced state between model accuracy and model complexity. Instead, our proposed method, which reconstructs the model architecture of the DETR, finds a balanced state between model accuracy and model complexity.

III. Methodology

The framework of our method is shown in Fig. 2. We propose a novel lightweight object detection framework based on the transformer. We employed the Swin Transformer to extract multiscale features and proposed utilizing an efficient HEFF feature fusion module to obtain fine-grained features directly. To speed up the convergence of the model, we used denoising training in the decoder. Moreover, this article introduced a pioneering exploration into the exclusion of the encoder structure of the DETR. The Swin Transformer with the HEFF module can theoretically assume the role of the encoder, extracting features suited for object detection. This elimination of the encoder structure can reduce the computational complexity of the model.

A. Multiscale Feature Extraction and Fusion

Carion et al. [10] proposed the first transformer-based detector, called the DETR, which provided a new paradigm for object detection. However, this approach suffers from poor detection performance, especially for small objects. The DETR employs ResNet [34] as its feature extraction backbone, which operates through continuous convolutional and pooling operations, where it progressively reduces the feature map size while increasing the number of channels to capture higher level feature information. However, ResNet uses a fixed receptive field, where the size and stride of the convolution kernel are fixed at each layer in the network. As the depth of the network increases, although the size of the feature map decreases and the receptive field increases to some extent with respect to the original input image, it is still limited, and it may be difficult to capture large-scale contextual information, especially for certain large-scale objects or scenes with broad backgrounds.

For the object detection task, a fixed receptive field may be a limiting factor because the object detection task usually requires the model to be able to simultaneously consider a larger range of contextual information in the input image to better understand the image content or accurately localize objects.

We utilize the Swin Transformer to extract multiscale features. As shown in Fig. 2, the image is input to the patch partition module to split into patches, and then, feature maps of different sizes are constructed in four stages. Specifically, the Swin Transformer is a patch merging operation similar to the pooling operation; it can merge neighboring small patches into a large patch and increase the receptive field. The patch merging operation is performed first between each stage to form a hierarchical transformer capable of extracting multiscale features and obtaining multiscale feature information. In addition, traditional transformers typically compute attention globally, while the

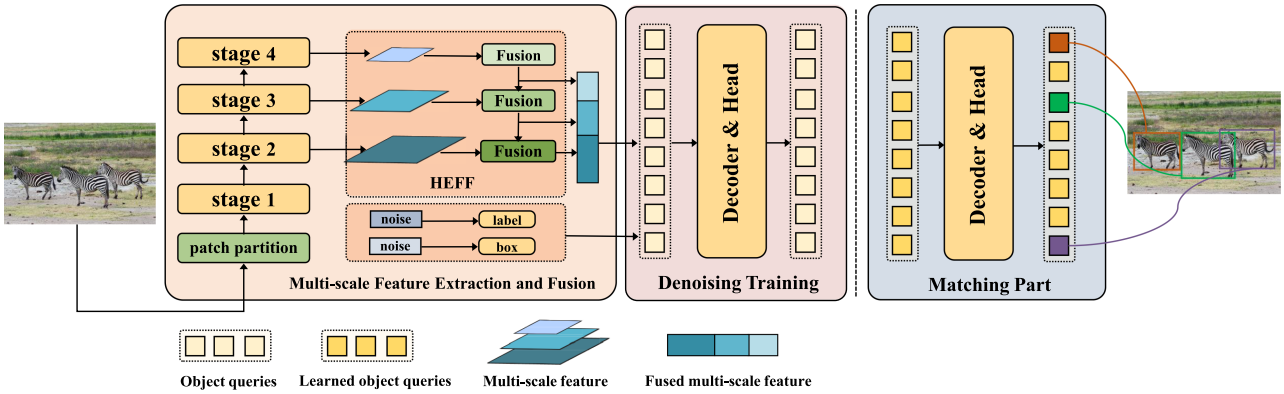

Fig. 2. Overall structure of the EF-DETR.

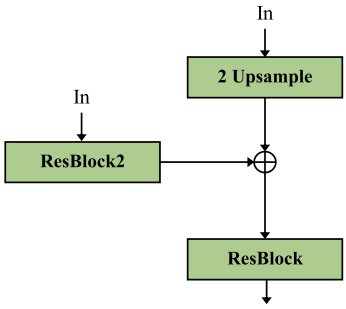

Fig. 3. HEFF fusion block.

Swin Transformer restricts attention computation within each window, thereby reducing the computational overhead, which is covered in more detail in Section III-B.

However, directly using the multiscale features extracted by the Swin Transformer as input to the subsequent decoder does not directly take full advantage of fine-grained features, which may not exist only at a single scale. Some fine-grained features may be more easily captured in higher level feature maps, while others may require lower level feature maps. To capture fine-grained features more comprehensively, features often need to be analyzed and integrated at multiple scales. Therefore, we propose HEFF, an efficient feature fusion module for fusing multiscale feature maps.

The images are input into the Swin Transformer to generate three features with different scales, and then, we input the list of multiscale tokens into the HEFF module. This module fuses all available multiscale tokens by using multiple fusion blocks before inputting the extracted multiscale tokens into the decoder, as shown in Fig. 3. This method extracts more effective complementary information from the feature mappings at different scales than do the previously developed simple linear aggregation methods.

More specifically, we gather the multiscale tokens extracted by the Swin Transformer to form a multiscale feature map and subsequently input them into the fusion module, which generates a feature pyramid through a top-down path among the features. The two input feature maps of this module are the high-resolution feature map x^l and the fused feature map x^{l+1}_{fuse} obtained by the previous fusion module

$$x^l_{\text{fuse}} = \text{ResBlock}\left(\text{Upsample}(x^{l+1}_{\text{fuse}}) + \text{ResBlock2}(x^l)\right) \quad (1)$$

where Upsample resizes the low-resolution feature map via bilinear interpolation to fuse it with the high-resolution feature map, ResBlock is the bottleneck residual block for feature smoothing and ResBlock2 is the bottleneck residual block for feature transformation. The result is a fusion of multiscale features, which are spliced together into an input for the decoder.

B. Encoder-Free Neck

Generally, the features extracted by a DETR-like model are then input into the transformer encoder to extract more suitable features for object detection. Specifically, the transformer encoder is utilized to learn global features, which helps during the subsequent detection task. However, the transformer encoder is accompanied by high computational and spatial complexity levels when computing the attention weights, which is squarely related to the number of feature pixels, resulting in the high computational complexity of DETR-like models.

In the previous section, we discussed our methodology of using a Swin Transformer for multiscale feature extraction. The Swin Transformer can efficiently extract global features, and with the help of the HEFF module, we can directly extract fine-grained features that are suitable for the object detection task.

Moreover, the Swin Transformer utilizes multihead self-attention based on shifted windows (SW-MSA), whereas the transformer encoder uses multihead self-attention (MSA) to perform global self-attention. The computational complexity of MSA is quadratic with respect to the input image size, whereas SW-MSA restricts the computation process to a fixed window size, significantly reducing the computational complexity of this approach. We assume that an image has dimensions of $h \times w$ and that each window contains $n \times n$ patches. The computational complexities of MSA and SW-MSA are formulated as follows:

$$\Omega(\text{MSA}) = 4\,hwC^2 + 2(hw)^2C \quad (2)$$

$$\Omega(\text{SW}-\text{MSA}) = 4\,hwC^2 + 2\,M^2hwC \quad (3)$$

where M represents the width and height of each window and C represents the depth of each pixel.

Therefore, we abandon the encoder structure, and we utilize the more lightweight Swin Transformer with HEFF for fine-grained feature extraction, which greatly reduces the computational complexity of the model.

C. Denoising Training

The DETR approaches object detection as a set prediction problem and utilizes the Hungarian matching algorithm to calculate object matches. However, the discrete nature of the Hungarian matching algorithm and the randomness of model training cause GT matching to become a dynamic and unstable process.

To address this problem, we must first understand the function of the decoder. In many works, the object queries in the decoder are considered 4-D coordinates (x, y, w, h), as understood by Liu et al. [33]. In each layer of the decoder, a relative offset $(\Delta x, \Delta y, \Delta w, \Delta h)$ is predicted, and the bounding box is updated to obtain a more accurate bounding box prediction (X, Y, W, H), as described in (4), which is then passed to the next layer

$$(X, Y, W, H) = (x, y, w, h) + (\Delta x, \Delta y, \Delta w, \Delta h). \quad (4)$$

We can think of the decoder as a module that learns two things: a 4-D coordinate (x, y, w, h) and the relative offset $(\Delta x, \Delta y, \Delta w, \Delta h)$. Due to the instability of the Hungarian matching algorithm mentioned previously, this can lead to the learning of unstable coordinates, which makes the learning of relative offsets difficult. Drawing upon insights from Li et al. [35] work, we use a denoising task as a shortcut to learn the relative offset, which skips the matching process and directly conducts learning. We consider the object queries to be 4-D coordinates. As described previously, we can add a small perturbation as noise near the GT so that our denoising task has the clear goal of directly reconstructing the GT without performing bipartite graph matching.

For each image, we add random noise to the bounding box and class label. The details are as follows:

Box noise: Box noise consists of two parts: center shifting and box scaling. The intensity of the shift and the size of the scaling are controlled by the hyperparameters λ_1 and λ_2. For center shifting, noise is added to make the center point of the real box undergo a small shift. Specifically, this random noise is denoted as $(\Delta x, \Delta y)$ and ensures that $|\Delta x| < \frac{\lambda_1 w}{2}$ and $|\Delta y| < \frac{\lambda_1 h}{2}$, such that the center of the noise frame still lies within the original bounding box. For box scaling, this means randomly scaling the length and width of the box, which is controlled by the λ_1 hyperparameter. Specifically, we randomly sample the box width and height by $[(1-\lambda_2)w, (1+\lambda_2)w]$ and $[(1-\lambda_2)h, (1+\lambda_2)h]$, respectively.

Label noise: We randomly flip the real label to any of the remaining labels with a certain probability. For example, if 80 categories are contained in the COCO dataset, we flip them to 80 other labels with a certain probability. The label-to-noise ratio to flip is controlled by the γ hyperparameter.

Algorithm 1: Algorithm for the EF-DETR.

Input: Images and their corresponding annotations.
Output: Position and label of the object in the image.

Let V_s, V_m, V_l denote the multiscale features extracted by the Swin Transformer, nq denote queries with noise, bn and ln denote box noise and label noise, P denote the input, $DT\ Decoder$ denote the decoder with denoising training, and FNN denote a feedforward neural network;
for *epoch*=1 **to** E **do**
  $V_s, V_m, V_l = Swin\ Transformer\,(P)$;
  $V_{L-1} = HEFF(V_s, V_m, V_l)$;
  $q = initialize\ query()$;
  $nq = prepare\ for\ nquery(bn, ln)$;
  $query = cat(q, nq)$;
  for *m*=1 **to** M **do**
    $X_L = DT\ Decoder\,(V_L, query)$;
  end
  $c, b = FNN(X_L)$;
  $closs, bloss = Loss(c, b, gt)$.
end

Notably, the denoising part is utilized only during the training period and is discarded during inference, where only the matching part is used.

The complete flow of the EF-DETR is shown in Algorithm 1. The body of our model consists of a Swin Transformer for extracting multiscale features and the HEFF fusion model for multiscale feature fusion. First, we input the given image P into the Swin Transformer to obtain its multiscale features V_s, V_m, V_l, and then, we input V_s, V_m, V_l into the HEFF module for adequate feature fusion to obtain the fused features V_{L-1}. Furthermore, we preprocess the noise query embedding nq with box noise and label noise, concatenate it with the normal query q and then input the concatenated query into the decoder to conduct denoising training together with the fused features V_L. Finally, the FNN is used to predict the position b of the object and its label c.

IV. Experiments

Section IV-A describes the dataset and implementation details used in this work. Then, to demonstrate the effectiveness of the proposed EF-DETR, Section IV-B compares representative and state-of-the-art methods and provides the corresponding analysis. In Section IV-D, ablation studies on the HEFF feature fusion module and hyperparameters in denoising training are described.

A. Dataset and Implementation Details

Dataset: We validate our approach on the COCO2017 dataset, which was developed by Microsoft and is known for its complex backgrounds, numerous objects, and smaller object sizes, making it a challenging benchmark. The performance on COCO is a strong indicator of a model's robustness and adaptability. In our experiments, we trained on train2017 and tested on

TABLE I
COMPARISON OF THE CURRENT OBJECT DETECTION MODELS ON THE COCO 2017 VALIDATION DATASET

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
CNN-based object detector							
Faster RCNN [17]	/	40.2	61.0	43.8	24.2	43.5	52.0
Mask RCNN [18]	/	39.8	62.3	43.4	22.1	43.2	51.2
Cascade RCNN [19]	/	42.8	62.1	46.3	23.7	45.5	55.2
RetinaNet [20]	/	39.1	59.1	42.3	21.8	42.7	50.2
Transformer-based object detector							
DETR [10]	500	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5 [10]	500	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101 [10]	500	43.5	63.8	46.4	21.9	48.0	61.8
UP-DETR [30]	300	42.8	63.0	45.3	20.8	47.1	61.7
Conditional DETR-R50 [31]	50	40.9	61.8	43.3	20.8	44.6	59.2
Conditional DETR-DC5-R50 [31]	50	43.8	64.4	46.7	24.0	47.6	60.7
Deformable DETR [29]	50	43.8	62.6	47.7	26.4	47.1	58.0
SMCA-R50 [32]	50	43.7	63.6	47.2	24.2	47.0	60.4
SMCA-R101 [32]	50	44.4	65.2	48.0	24.3	48.5	61.0
DAB-DETR-DC5-R50 [33]	50	44.5	65.1	47.7	25.3	48.2	62.3
DAB-DETR-DC5-R101 [33]	50	45.8	65.9	49.3	27.0	**49.8**	63.8
EF-DETR (ours)	50	**48.3**	**67.7**	**51.9**	**29.7**	**51.6**	**66.0**

val2017, reporting the average precision (AP) values at various IoU thresholds and object scales, following standard practices.

Implementation details: In our experiments, the model was implemented using PyTorch and trained on an NVIDIA 3090 GPU with 24 GB of memory. Regarding the hyperparameters, we adopted the same approach as the DETR, utilizing a five-layer transformer decoder with 256 hidden dimensions. For the learning rate scheduler, we used an initial learning rate (lr) of 10^{-4}, and we reduced the lr at the 40th epoch by setting a multiplication factor of 0.1 for 50 epochs and at the 11th epoch by setting a multiplication factor of 0.1. We used the hyperparameters associated with the noise to be $\lambda_1 = 0.4$, $\lambda_2 = 0.4$ and $\gamma = 0.2$, and we used the weighted Adam (AdamW) optimizer with a weight decay of 10^{-4}. The batch size was 2.

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

To verify the validity of our proposed model, we selected several representative object detection methods, including the classic CNN-based detector and transformer-based object detectors, such as the DETR, UP-DETR, conditional DETR, deformable DETR, SMCA, and DAB-DETR. "DC5" in DETR-DC5 means that a dilated convolution was added to the last stage of the backbone network and that the pooling layer was reduced to double the resolution. R50 and R101 represent ResNet-50 and ResNet-101, respectively. The conditional DETR, SMCA, deformable DETR, and DAB-DETR are recent models with strong performance. The experimental results on the COCO dataset are presented in Table I, where our model is compared with others in relation to the number of training epochs and the detection accuracy.

The table illustrates the variations in the convergence behavior of these models, with the DETR needing 500 epochs, UP-DETR requiring 300 epochs, and the EF-DETR requiring only 50 epochs to reach convergence because we use denoising training in the decoder.

In terms of the detection accuracy, the proposed EF-DETR achieved a high precision of 48.3 on the COCO2017 validation dataset. Compared to the baseline DETR model, the detection accuracy for each object size has improved, with the highest increase seen in the detection accuracy for small objects AP_s, which increased by 9.2. This improvement can be attributed to the use of the multiscale feature extractor Swin Transformer and the multiscale feature fusion, HEFF. By using the multiscale feature extractor Swin Transformer, the model is able to extract feature information at multiple scales, especially for small objects whose features may not be obvious enough at a single scale in the image due to factors such as their small size or occlusion. The combined features at multiple scales can better capture their feature information, which improves the small object detection accuracy. The HEFF module is capable of fine-grained feature fusion, and since fine-grained features may be distributed at different scales, the HEFF module is able to analyze and integrate features at multiple scales. Through this fusion, the model is able to better understand the image content, which results in better performance during the object detection task.

In addition, to more fully evaluate the generalization performance of the proposed method, we conducted additional supplementary experiments on the VisDrone dataset, which is an object detection dataset used for UAVs in which the density of objects is high and most of them are small (<32 pixels). As shown in Table II, compared to other transformer-based models, the overall performance of our model is superior, which indicates that our proposed method performs better on small objects, thanks to the multiscale feature extractor we propose to use and the HEFF module.

C. Computational Complexity Analysis

To demonstrate the balance between model accuracy and computational complexity achieved by our proposed method, we compared our method with our baseline DETR and two other advanced models, the conditional DETR and DAB-DETR, considering the parameters, GFLOPs, and detection accuracy.

TABLE II
COMPARISON OF THE CURRENT OBJECT DETECTION MODELS ON THE VISDRONE VALIDATION DATASET

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR [10]	8.3	18.3	6.5	2.2	13.5	25.9
UP-DETR [30]	8.1	18.6	7.0	2.1	14.9	25.1
Conditional DETR [31]	9.0	19.1	7.6	3.8	15.7	24.3
Deformable DETR [29]	8.6	18.6	7.2	5.7	15.5	24.1
EF-DETR (Ours)	**13.2**(+4.9)	**24.2**(+5.9)	**12.3**(+5.8)	**8.1**(+5.9)	**18.4**(+4.9)	**35.7**(+9.8)

TABLE III
COMPUTATIONAL COMPLEXITY ANALYSIS

Model	Params(M)	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR [10]	41	86	42.0	62.4	44.2	20.5	45.8	61.1
Conditional DETR-R101 [31]	63	156	42.8	63.7	46.0	21.7	46.6	60.9
DAB-DETR-R101 [33]	63	174	43.5	63.9	46.6	23.6	47.3	61.5
Conditional DETR-DC5-R101 [31]	63	262	45.0	65.5	48.4	26.1	48.9	62.8
DAB-DETR-DC5-R101 [33]	63	282	45.8	65.9	49.3	27.0	49.8	63.8
EF-DETR (Ours)	**46**	**67**	**48.3**	**67.7**	**51.9**	**29.7**	**51.6**	**66.0**

Table III indicates that as the detection accuracy AP increases, the number of GFLOPs also increases, implying an increase in the computational complexity of the models. The tradeoff between model complexity and accuracy is not favorable for industrial facilities with limited computing resources. However, our proposed EF-DETR achieves an AP of 48.3 with only 67 GFLOPs, improves the AP by 15% and reduces the GFLOPs by 22.1% compared to the DETR. This result is attributed to the fact that we removed the transformer encoder structure, which has a high computational complexity, from the DETR. Furthermore, since the Swin Transformer is a transformer encoder structure, we explored the possibility of directly adopting the Swin Transformer as the backbone since it is more efficient when handling large-scale data due to its hierarchical feature representation and window multihead self-attention mechanism.

From the experimental results, it can be seen that our method not only simplifies the model structure but also improves the detection accuracy; it strikes a balance between computational complexity and performance and provides a more feasible solution for practical application of the object detection task.

D. Ablation Experiments

In the ablation analysis, we explore the impact of the HEFF module on the model performance and the effect of hyperparameters on the experimental performance.

Importance of HEFF: To demonstrate the effectiveness of the feature fusion component, we retain the other model structures while removing the HEFF module and utilize ResNet-50 for feature extraction.

As shown in Table IV, the module adds only 1 M parameters but significantly improves the detection accuracy. After the HEFF module is removed, the detection accuracies achieved for all categories decrease, where AP_L decreases by 3.6, AP_M decreases by 1.6, AP_S decreases by 0.9, AP_{75} decreases by 1.5, AP_{50} decreases by 1.9, and AP decreases by 1.5. These results

TABLE IV
ABLATION EXPERIMENT FOR THE FEATURE EXTRACTION METHOD

HEFF	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
	46.8	65.8	50.4	28.8	50.2	62.4
✓	**48.3** (+1.5)	**67.7** (+1.9)	**51.9** (+1.5)	**29.7** (+0.9)	**51.6** (+1.6)	**66.0** (+3.6)

TABLE V
SENSITIVE PARAMETER EXPERIMENTS

λ_1/λ_2	γ	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0.2	0.2	45.4	64.3	48.7	27.9	49.9	63.9
0.4	**0.2**	**48.3**	**67.7**	**51.9**	**29.7**	**51.6**	**66.0**
0.2	0.4	46.9	65.4	50.2	28.7	49.8	64.6
0.4	0.4	46.1	64.5	49.8	28.1	49.2	64.2

indicate that the HEFF module significantly enhances the fusion of multiscale features, thereby improving the overall detection accuracy of the model. Moreover, this module boosts the model's detection performance without augmenting its computational load, which is achieved simply through a straightforward adjustment of network connections.

Hyperparameter sensitivity experiment: To determine the optimal hyperparameter combination that enhances model performance and generalizability, we conducted sensitivity experiments on the box noise hyperparameters λ_1 and λ_2 and label noise hyperparameters γ. We setup four different value combinations, keeping the other parameters consistent for training, and tested the detection accuracy of the model, as shown in Table V. It is evident that the model achieves the best accuracy when $\lambda_1 = 0.4$, $\lambda_1 = 0.4$, and $\gamma = 0.2$.

In addition, we also conducted sensitivity experiments on the decoder layers' hyperparameters l. We experimented with l using three, four, five, and six decoder layers to find the most appropriate number, as shown in Table VI. It is evident that the model achieves the best accuracy when $l = 5$.

TABLE VI
SENSITIVE PARAMETER EXPERIMENTS

l	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
3	46.6	65.4	49.8	28.7	50.1	63.4
4	46.4	65.5	50.0	28.4	49.6	62.9
5	**48.3**	**67.7**	**51.9**	**29.7**	**51.6**	**66.0**
6	47.5	67.2	51.0	28.9	51.0	64.8

V. CONCLUSION AND DISCUSSION

The main contribution of this article is the proposal of a novel lightweight object detection framework based on the transformer architecture. We introduced the use of a multiscale feature extractor, the Swin Transformer, and HEFF to obtain fine-grained features and employed denoising training in the decoder to increase the convergence speed of the model. Furthermore, this article suggests abandoning the DETR encoder structure, as the Swin Transformer with the HEFF module can directly replace the encoder. This abandonment of the structure provides a new approach for the development and further optimization of lightweight object detection networks. The experimental results in Section IV demonstrate that our proposed model has fewer parameters and fewer GFLOPs than mainstream transformer-based detectors, yet it achieves outstanding accuracy. It is suitable for deployment in industrial facilities with limited computational power. In the future, we plan to extend the proposed approach to more specific tasks, such as the identification, tracking and classification of industrially produced goods, to enhance the process of storing, loading and distributing goods. However, there is still room for improvement in the approach proposed in this article. Therefore, we will further optimize the performance and efficiency of the model for practical applications in different industrial facilities. In addition, we will continue to explore lightweight improvements of the model to reduce its computational and storage costs and increase the deployment efficiency in resource-constrained environments. We believe that these improvements will further enhance the utility and reliability of our proposed approach in the industrial domain.

REFERENCES

[1] X. Shen, J. Liu, J. Sun, L. Jiang, H. Zhao, and H. Zhang, "SSCT-Net: A semisupervised circular teacher network for defect detection with limited labeled multiview MFL samples," *IEEE Trans. Ind. Inform.*, vol. 19, no. 10, pp. 10114–10124, Oct. 2023.

[2] Z. Geng, C. Shi, and Y. Han, "Intelligent small sample defect detection of water walls in power plants using novel deep learning integrating deep convolutional GAN," *IEEE Trans. Ind. Inform.*, vol. 19, no. 6, pp. 7489–7497, Jun. 2023.

[3] M. Wieczorek, J. Silka, M. Wozniak, S. Garg, and M. M. Hassan, "Lightweight convolutional neural network model for human face detection in risk situations," *IEEE Trans. Ind. Inform.*, vol. 18, no. 7, pp. 4820–4829, Jul. 2022.

[4] M. Wang, L. Zhao, and Y. Yue, "PA3DNet: 3-D vehicle detection with pseudo shape segmentation and adaptive camera-LiDAR fusion," *IEEE Trans. Ind. Inform.*, vol. 19, no. 11, pp. 10693–10703, Nov. 2023.

[5] Y. Zhu, C. Chen, G. Yan, Y. Guo, and Y. Dong, "AR-Net: Adaptive attention and residual refinement network for copy-move forgery detection," *IEEE Trans. Ind. Inform.*, vol. 16, no. 10, pp. 6714–6723, Oct. 2020.

[6] A. Shahbaz and K. Jo, "Deep atrous spatial features-based supervised foreground detection algorithm for industrial surveillance systems," *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4818–4826, Jul. 2021.

[7] C. Zhang, K.-M. Lam, and Q. Wang, "CoF-Net: A progressive coarse-to-fine framework for object detection in remote-sensing imagery," *IEEE Trans. Geosci. Remote. Sens.*, vol. 61, 2023, Art. no. 5600617.

[8] C. Zhang, J. Su, Y. Ju, K.-M. Lam, and Q. Wang, "Efficient inductive vision transformer for oriented object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote. Sens.*, vol. 61, 2023, Art. no. 5616320.

[9] C. Zhang, T. Liu, J. Xiao, K.-M. Lam, and Q. Wang, "Boosting object detectors via strong-classification weak-localization pretraining in remote sensing imagery," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 5026520.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 84–90.

[13] J. Hou, C. Luo, F. Qin, Y. Shao, and X. Chen, "FUS-GCN: Efficient B-rep based graph convolutional networks for 3D-CAD model classification and retrieval," *Adv. Eng. Inform.*, vol. 56, Apr. 2023, Art. no. 102008.

[14] F. Qin, N. Gao, Y. Peng, Z. Wu, S. Shen, and A. Grudtsin, "Fine-grained leukocyte classification with deep residual learning for microscopic images," *Comput. Methods Programs Biomed.*, vol. 162, pp. 243–252, Aug. 2018.

[15] Y. Chen et al., "Deep object detection network based automatic spinopelvic parameters measurement method for adult spinal deformity classification," in *Proc. 4th Int. Seminar Artif. Intell., Netw. Inf. Technol.*, 2023, pp. 471–478.

[16] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 9243–9275, Aug. 2022.

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

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

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

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

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

[22] X. Liang, W. Cheng, C. Zhang, L. Wang, X. Yan, and Q. Chen, "YOLOD: A task decoupled network based on YOLOv5," *IEEE Trans. Consum. Electon.*, vol. 69, no. 4, pp. 775–785, Nov. 2023.

[23] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3285–3294.

[24] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Representations*, 2020, pp. 1–22.

[25] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 548–558.

[26] W. Tang, F. He, Y. Liu, and Y. Duan, "MATR: Multimodal medical image fusion via multiscale adaptive transformer," *IEEE Trans. Image Process.*, vol. 31, pp. 5134–5149, 2022.

[27] M. Kaselimi, A. Voulodimos, I. Daskalopoulos, N. Doulamis, and A. Doulamis, "A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3299–3307, Jul. 2023.

[28] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.

[29] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. 9th Int. Conf. Learn. Representations*, 2021, pp. 1–16.

[30] Z. Dai, B. Cai, Y. Lin, and J. Chen, "UP-DETR: Unsupervised pre-training for object detection with transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern. Recognit.*, 2021, pp. 1601–1610.
[31] D. Meng et al., "Conditional DETR for fast training convergence," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3631–3640.
[32] P. Gao, M. Zheng, X. Wang, J. Dai, and H. Li, "Fast convergence of DETR with spatially modulated co-attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 3601–3610.
[33] S. Liu et al., "DAB-DETR: Dynamic anchor boxes are better queries for DETR," in *Proc. 10th Int. Conf. Learn. Representations*, 2022, pp. 1–19.
[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern. Recognit.*, 2016, pp. 770–778.
[35] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern. Recognit.*, 2022, pp. 13609–13617.

Siyi Cheng received the bachelor's degree in software engineering from the School of Software, Nanchang University, Nanchang, China, in 2022. She is currently working toward the master's degree in electronic information with the School of Computer Science and Technology, Chongqing University, Chongqing, China.

She is currently engaged in the research of computer vision, object detection, camouflage object detection, and object tracking.

Jingnan Song received the bachelor's degree in computer science and technology from the College of Information Technology, Hebei University of Economics and Business, Hebei, China, in 2021. She is currently working toward the master's degree in electronic information with the School of Computer Science and Technology, Chongqing University, Chongqing, China.

She is currently engaged in the research of computer vision, object detection, and object tracking. Her master's thesis is the research of weak and small object detection and tracking algorithm in complex low-altitude background environments.

Mingliang Zhou received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2017.

He is currently an Associate Professor with the School of Computer Science, Chongqing University, Chongqing, China. From 2017 to 2019, he was a Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong. From 2019 to 2021, he was a Postdoctoral Fellow with the State Key Lab of Internet of Things for Smart City, University of Macau, Macau, China. His research interests include image and video coding, perceptual image processing, multimedia signal processing, rate control, multimedia communication, machine learning, and optimization.

Xuekai Wei received the bachelor's degree in electronic information science and technology and the master's degree in communication and information systems from Shandong University, Jinan, China, in 2014 and 2017, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2021.

He is currently an Associate Professor with the School of Computer Science, Chongqing University, Chongqing, China. He was a Postdoctoral Fellow with the School of Artificial Intelligence, Beijing Normal University, Beijing, China. His research interests include video coding/transmission and machine learning. Focusing on network-centric transmission and video-centric encoding problems, his research strives to close the gap between compression and delivery through joint optimization methods incorporating knowledge of content characteristics and link capabilities.

Huayan Pu received the M.Sc. and Ph.D. degrees in mechatronics engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2011, respectively.

She is currently a Professor with the State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing, China. Her research interests include vibration controlling and robotics.

Dr. Pu was a recipient of the Best Paper Award in biomimetics at the 2013 IEEE International Conference on Robotics and Biomimetics. She was also nominated as the Best Conference Paper Finalist at the 2012 IEEE International Conference on Robotics and Biomimetics.

Jun Luo received the B.S. and M.S. degrees in mechanical engineering from Henan Polytechnic University, Jiaozuo, China, in 1994 and 1997, respectively, and the Ph.D. degree in mechatronics engineering from the Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, China, in 2000.

He is currently a Professor with the State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing, China. His research interests include artificial intelligence, sensing technology, and special robotics.

Weijia Jia (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in computer science from Central South University, Changsha, China, in 1982 and 1984, respectively, and the M.A.Sc. and Ph.D. degrees in computer science from the Polytechnic Faculty of Mons, Hainaut, Belgium, in 1992 and 1993, respectively.

He is currently the Chair Professor and Director of the BNU-UIC Institute of Artificial Intelligence and Future Networks, Zhuhai, China, and the VP for Research of BNUHKBU United International College, Zhuhai China. He has authored or coauthored more than 500 publications in prestigious international journals/conferences and research books and book chapters. His research interests include smart city, IoT, knowledge graph constructions, multicast and anycast QoS routing protocols, wireless sensor networks, and distributed systems.