IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 January 2024, accepted 3 February 2024, date of publication 6 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3363442

RESEARCH ARTICLE

MilDetr: Detection Transformer for Military Camouflaged Target Detection

BING LI, RONGQIAN ZHOU, LU YANG, QIWEN WANG, AND HUANG CHEN
School of Engineering, Shantou University, Shantou 515041, China

Corresponding author: Huang Chen (18850067761@163.com)

This work was supported by the Start-Up Research Projects of Shantou University under Grant 140/09419026.

ABSTRACT Military Camouflage Target Detection (MCTD) is a special object detection task that aims to detect military camouflaged targets in the wild. In the challenging MCTD task, the confusing appearance and contours of military camouflaged targets often lead to the poor performance of existing methods. In this study, we propose an end-to-end Military Detection Transformer (MilDetr) for MCTD. We introduce two improvements to enhance the model's performance. First, we employ the Reverse Features Feed Forward Neural Network (R3FN) for local information aggregation in the encoder of MilDetr. In addition, the Fusion Previous Query (FPQ) module is utilized for multi-stage query feature fusion in the decoder of MilDetr. To overcome data limitations for MCTD, we build two simulation military camouflaged target datasets called MilDet and MilCls. The ablation experiments on MilDet reveal the effectiveness of our improvements. Experimental results demonstrate that MilDetr obtains 95.6 AP on MilDet. Furthermore, MilDetr obtains 96.4 AP on MilDet with the pre-trained weights on ImageNet and MilCls. Compared with other object detectors, MilDetr achieves end-to-end military camouflaged target detection with superior performance.

INDEX TERMS Military camouflaged target, object detection, deep learning.

I. INTRODUCTION

Object detection is an important and challenging task in computer vision, which aims to identify and localize objects in images. As a result, object detection has found widespread use in practical applications such as visual surveillance [1], human-machine interaction [2], and autonomous driving [3].

Military camouflage target detection (MCTD) aims at the detection of military camouflaged targets in the wild. It is a special object detection task. Military target camouflage [4] is the alteration of the outline and appearance of a target. Using colors and patterns similar to those of the environment, the targets are more difficult to detect or hit by military equipment [5]. This technology has a long history and has evolved in response to developments in technology and warfare. Military target camouflage allows soldiers to increase their survivability and mission effectiveness by preventing visual detection by other military equipment [6].

The performance of object detectors has been significantly enhanced with the growth of datasets and the rapid development of deep learning. Two typical architectures for object detectors are the CNN-based object detectors and the Transformer-based object detectors. CNN-based object detection models can be classified into two-stage models and one-stage models. Classical two-stage models include RCNN [7] and subsequent improvements such as Fast-RCNN [8], Faster-RCNN [9], Mask-RCNN [10], and Cascade-RCNN [11]. One-stage models include SSD [12], RetinaNet [13], and YOLO series [14], [15], [16], [17], [18], [19]. CNN-based object detectors use hand-crafted components, such as anchor generation and non-maximum suppression (NMS), which means they cannot be considered full end-to-end detectors. In 2020, the DEtection TRansformer (DETR) [20] was proposed as a successful application of the Transformer for object detection. Recently, several Transformer-based detectors have been proposed, including Conditional DETR [21], DAB-DETR [22], Deformable DETR [23], DN-DETR [24], and DINO [25]. Transformer-based object detectors eliminate the need for hand-crafted components, simplifying the object detection pipeline and achieving full end-to-end object detection. Transformer-based object detectors achieve comparable performance to

The associate editor coordinating the review of this manuscript and approving it for publication was Gianluigi Ciocca.

CNN-based object detectors. Therefore, Transformer-based object detectors are a recent area of focus in object detection methods.

MCTD is more difficult due to the confusing appearances and contours of the military camouflaged targets. MCTD has been researched since the 1950s [26], and there are six stages in the development of MCTD: statistical pattern recognition, knowledge-based object detection, model-based object detection, multi-modal information fusion methods, hybrid systems of artificial neural networks and expert systems, and deep learning methods. Multi-modal information fusion methods, such as the use of infrared imaging, can effectively detect camouflaged tanks, armored vehicles, and other targets with heat sources [27], [28]. However, the cost of infrared imagers is significantly higher than that of conventional cameras. In addition, camouflaged targets will try to avoid emitting heat signals to avoid being detected. Therefore, the detection of camouflaged targets using RGB images captured by conventional cameras is a current research focus. In 1998, Tankus et al. [29] proposed a non-edge-based mechanism for the detection of regions of interest for detecting camouflaged targets in both natural environments and battlefields. Since then, many researchers have started to use human visual features (e.g., color, texture, optical flow, etc.) to describe camouflaged targets and proposed the feature-based camouflaged target detectors [30], [31], [32], [33]. Compared with object detection, there are fewer deep learning-based methods for MCTD due to the limited data available. Some CNN-based detectors have been improved and introduced to MCTD. Recently, Bowen Yu [34] introduced an improved YOLOv3 for detecting military targets, while Deng et al. [35] introduced an improved RetinaNet with attention mechanisms for detecting camouflaged people. However, the well-hidden targets often lead to the poor performance of existing methods. In addition, the above methods do not achieve full end-to-end military camouflaged target detection. Our research reveals that no transformer-based object detector has been applied to MCTD. Therefore, there is still potential for further exploration of the end-to-end Transformer-based object detectors for MCTD.

To find better military camouflaged target detection methods, our research is focused on exploring end-to-end object detectors and overcoming data limitations for MCTD. We generate available military camouflaged target data and propose a novel end-to-end military camouflaged target detector called the Military Detection Transformer (MilDetr). The overview of the proposed MilDetr is shown in Fig. 1. In summary, the main contributions of this paper are threefold.

1) We propose an end-to-end Transformer-based object detector called Military Detection Transformer (MilDetr) for MCTD. MilDetr includes our novel Reverse Features Feed Forward Neural Network (R3FN) and Fusion Previous Query (FPQ) module to enable better military camouflaged target detection performance than existing object detectors. In the encoder of MilDetr, the proposed R3FN reintroduces local information by convolutional blocks after MSDA. In the decoder of MilDetr, the proposed FPQ fuses multi-layer queries by Geometric Sequence Sum Fusion (GSSF) and Fusion Gradient Truncation (FGT) approaches.
2) We build two simulation military camouflaged target datasets called MilDet and MilCls.
3) The ablation experiments show that the proposed methods improve the performance of military camouflaged target detection. The comparison experiments indicate that MilDetr achieves state-of-the-art (SOTA) performance on the MilDet by training with the Contrastive DeNoising (CDN) strategy.

II. RELATED WORKS

A. DETR

Proposed by Facebook AI Research in 2020, DETR is a Transformer-based object detector that analyzes the relationship between objects and global information with learnable object queries. Compared to common CNN-based object detection methods, DETR predicts object classes and position boxes directly from the whole image without generating candidate boxes. Specifically, DETR treats object detection as an ensemble prediction problem, which means that the targets in an image are treated as an ensemble rather than a series of discrete bounding boxes. DETR allows the position and size variations of overlapping objects to be better handled. The NMS process is avoided, and the object detection pipeline is greatly simplified.

As shown in Fig. 2, DETR uses a CNN backbone to learn a 2D representation of the input image and flattens the 2D representation with spatial positional encoding added. Then a Transformer encoder extracts features and computes global information through Multi-head Self-Attention (MHSA) blocks. A Transformer decoder then considers a fixed number N_{num} of learned positional embeddings called object queries, and the output of the encoder, generating N_{num} decoder output embeddings. Each output embedding is fed into a shared prediction feed-forward network (FFN) and gives a prediction. During the training stage, DETR uses the Hungarian matching algorithm to match the prediction bounding boxes and the ground truths and calculates the bipartite matching loss based on the matching results.

DETR lays the foundation for the subsequent Transformer object detection framework. However, the framework faces challenges such as expensive training costs, high dependence on large datasets, slow convergence speed, and poor performance on small objects.

B. MULTI-SCALE DEFORMABLE ATTENTION

To address the problem of slow convergence speed and poor small object detection performance of DETR, Deformable DETR is proposed with the Multi-scale Deformable Attention (MSDA) mechanism to speed up the model convergence

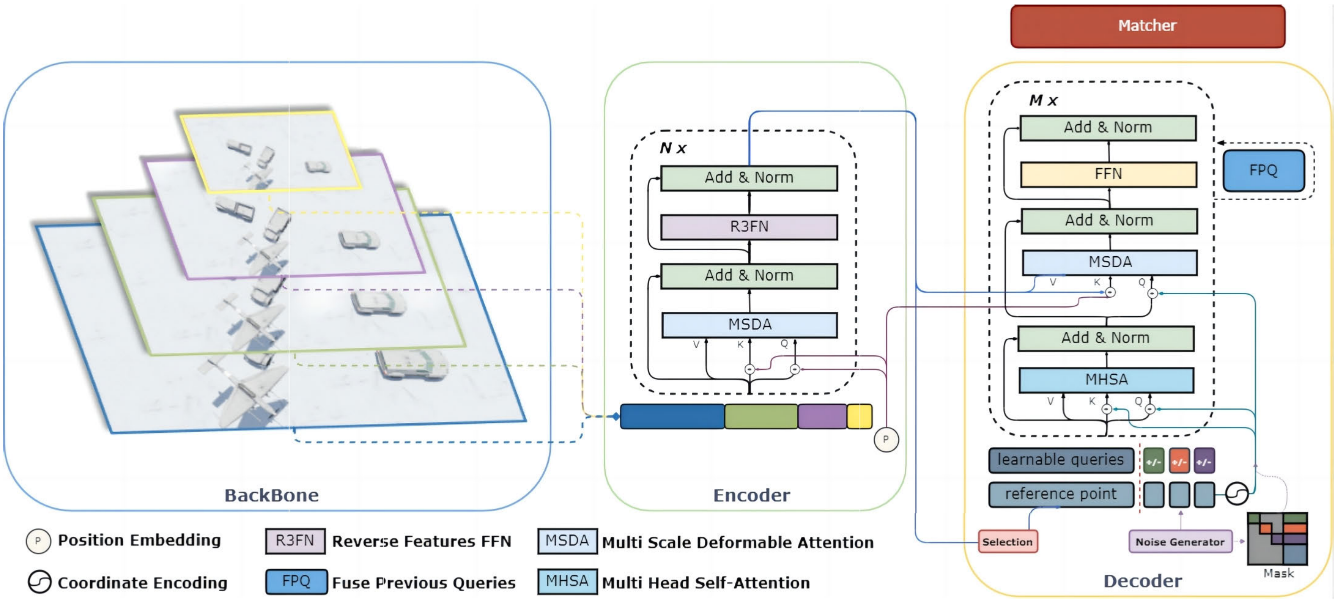


FIGURE 1. Overview of MilDetr.

and introduce the multi-scale features into the encoder for attention computation. Deformable DETR claims that no matter how large the spatial size of the feature map, the model only needs to focus on a small set of sample points around the reference point. As shown in Fig. 3, MSDA applies the Multi-head Deformable Attention module to multi-scale feature maps to collect multi-scale information with low computational cost.

The Deformable Attention module takes into account the deformations of objects at different scales when calculating attention weights. Specifically, the Deformable Attention module uses a deformable convolutional layer to generate deformation parameters that are used to adjust the shape and size of the active area. Given input multi-scale feature maps $\{x^l\}_{l=1}^{L}$, where $x \in R^{C\times H\times W}$ and L is the total number of feature maps. Given K sampled points, the multi-scale deformable attention can be expressed in Equation (1):

$$MSDA(z_q, \hat{p}_q, \left\{x^l\right\}_{l=1}^{L}) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L} \sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l (\phi_l \left(\hat{p_q}\right) + \Delta p_{mqlk}) \right] \quad (1)$$

where p_q is the initial sample point, m is the index of the attention head, and k is the index of the sampled key. Δp_{mlqk} and A_{mlqk} denote the sampling offset and attention weight of the k^{th} sampling point in the l^{th} feature map in the m^{th} attention head. The attention weights A_{mlqk} are normalized, and the function $\phi_l \left(\hat{p_q}\right)$ serves to remap the normalized coordinates $\hat{p}_q$ to the l^{th} input feature map.

C. CONTRASTIVE DENOISING TRAINING APPROACH

To speed up the training process of DETR and address the unclear meaning of the object queries, DN-DETR effectively mitigates the problem of unstable bipartite graph matching by introducing the DeNoising (DN) training strategy. DINO proposes a Contrastive DeNoising (CDN) training strategy. This strategy feeds the model with positive and negative samples generated from the real images by a noise generator during the training process.

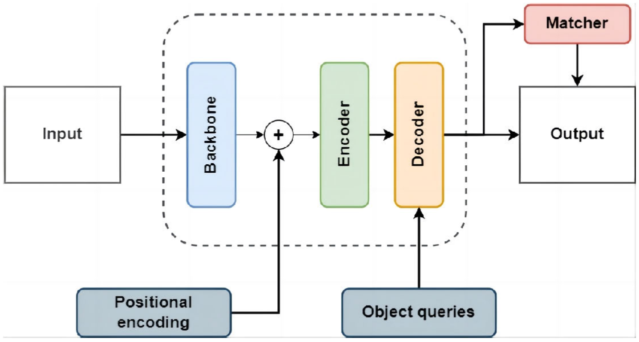


FIGURE 2. The framework of DEtection TRansformer.

As shown in Fig. 4, the noise generator has two hyper-parameters λ_1 and λ_2 ($\lambda_1 < \lambda_2$). Positive samples in the inner square have less noise than λ_1 and are used to reconstruct the corresponding positive samples. Negative samples generated between the internal and external square have noise scales greater than λ_1 and less than λ_2 and are expected to be in the "no object" class. Negative samples are useful for suppressing duplicate boxes and stable bipartite graph matching. When a smaller λ_2 is used, the negative samples are more similar to the ground truth boxes, and suppressing these negative samples can improve the model performance effectively. In this way, all ground truth boxes have multiple sets of positive and negative samples, which are used to improve the efficiency of denoising.

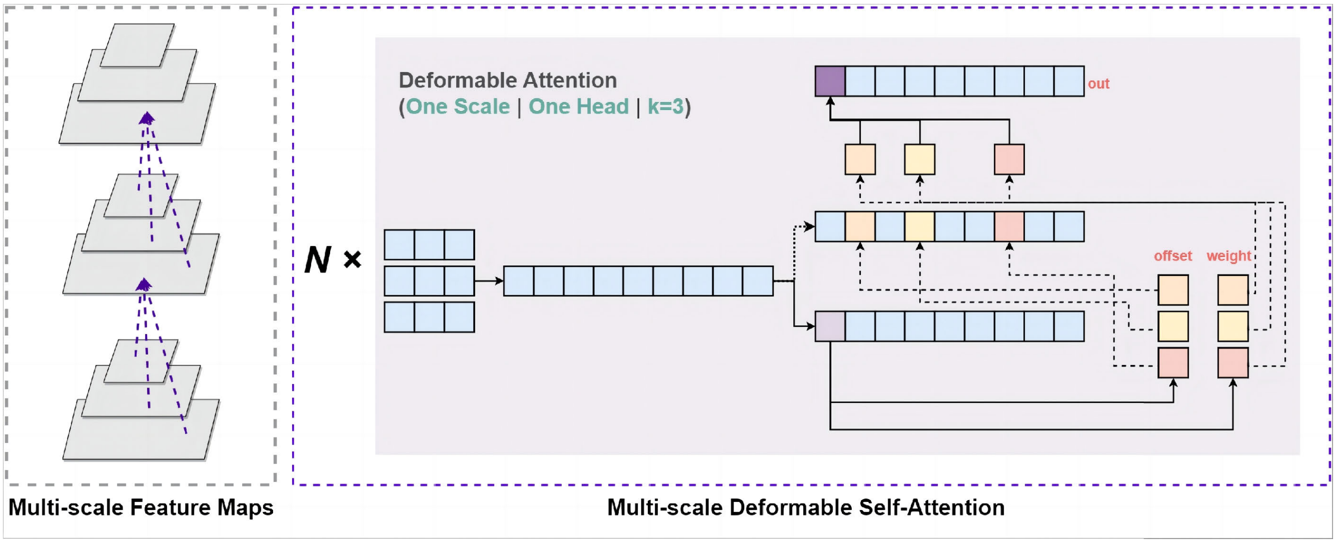

Deformable Attention
(One Scale | One Head | k=3)
out
N ×
offset
weight
Multi-scale Feature Maps
Multi-scale Deformable Self-Attention

FIGURE 3. The structure of multi-scale deformable attention mechanism.

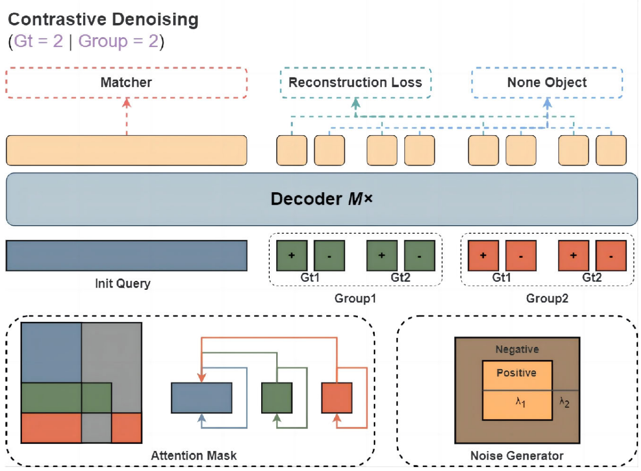


FIGURE 4. The structure of CDN group and demonstration of attention mask and noise generator.

III. METHODS

A. OVERVIEW OF MILDETR

As shown in Fig. 1, MilDetr is an end-to-end object detection framework consisting of a CNN backbone for generating image features, a multi-layer Transformer encoder for encoding the image features, a multi-layer Transformer decoder for decoding, and the multiple prediction heads to generate the object predict position box and class label.

Given an RGB image $x \in R^{3\times H\times W}$, MilDetr extracts corresponding multi-scale feature maps $\{x^l\}_{l=1}^{L}$ through the CNN backbone, and then feeds them into the encoder along with the corresponding position embedding. MilDetr uses the ImageNet [36] pre-trained ResNet50 [37] as the backbone and generates feature maps at four scales. Following the setting of Deformable DETR, the encoder of MilDetr has N encoder layers ($N = 6$), and the fixed positional encodings are added to the input of each attention layer. Moreover, each encoder layer consists of an MSDA module and a proposed Reverse Features Feed Forward Neural Network (R3FN). The output of the last encoder layer will be used as the key and value for cross-attention computation in the decoder. The decoder of MilDetr has M decoder layers ($M = 6$), and we introduce a novel feature fusion module named Fusion Previous Query (FPQ) in the decoder. MilDetr uses the CDN training strategy by taking the generated positive samples and negative noise samples as the input query of the Decoder. Finally, the outputs of all decoder layers are passed to the independent prediction heads to generate the prediction boxes and the prediction labels. The prediction results are then matched with the ground truth by the Hungarian algorithm for loss calculation.

B. R3FN

In object detection tasks, the targets often exist in a specific environment or coexist with other targets. Therefore, the target's neighborhood region can provide effective contextual information to help detect it. Many Transformer-based object detectors that focus on global information, such as Deformable DETR, show good performance in object detection. Using MSDA in the encoder layers helps Deformable DETR to converge faster and improves the performance of detecting irregular objects. However, we analyze the characteristics of camouflaged targets and highlight the need for object detectors to consider both local and global information in MCTD. Fig. 5 displays the results of the attention visualization from the Encoder of DETR and Deformable DETR. Both models focus on the unimportant edge regions of the images. The attention mechanism in the Transformer-based object detectors can be expressed in Equation (2). The attention mechanism takes a series of sampled pixels as input and transforms them into queries ($Q \in R^{(n+1)\times d}$), keys ($K \in R^{(n+1)\times d}$) and values ($V \in R^{(n+1)\times d}$), where n is the sequence length of the tokens and d is the embedding dimensions. Based on

Equation (2), the attention mechanism allocates attention based on feature similarity. We observe that the camouflaged targets blend in with the background due to their similar appearance. The visual features of the background regions are very similar to some regions in the camouflaged targets. Therefore, the camouflaged targets have a misleading nature for attention computation, and the attention mechanism may misallocate attention to a feature-similar background pixel that is unimportant for a camouflaged target. In addition, we find that Deformable DETR places more emphasis on the edge region of the input images. We hypothesize that MSDA makes the receptive field of each reference point in the query more sparse. The sampled pixels of each reference point in MSDA may come from different areas at different scales, which can lead to the lack of local information, causing the omission of well-camouflaged targets and the misdetection of background. Therefore, we suggest that the Transformer-based camouflaged object detectors should aggregate more local information.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}}V) \tag{2}$$

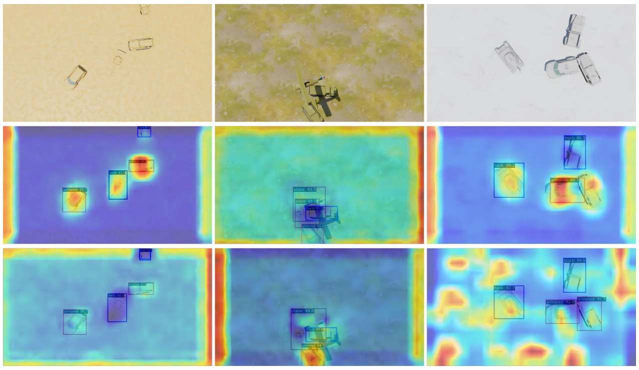


FIGURE 5. Input images (row 1), visualization of the attention maps from DETR (row 2), and visualization of the attention maps from Deformable DETR (row 3).

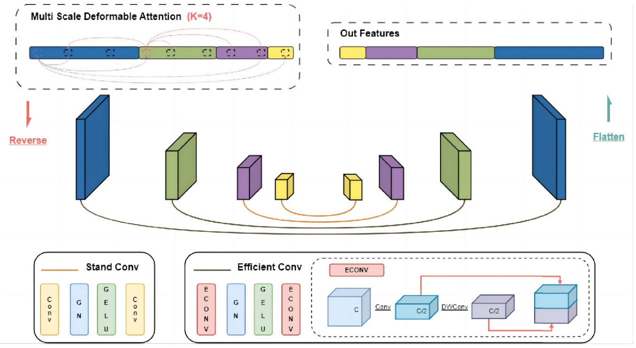


FIGURE 6. The structure of R3FN.

To mitigate the negative effects of MSDA, we reintroduce local information for each feature map using the Reverse Features Feed Forward Network (R3FN) module. Specifically, the R3FN module aggregates local information by performing a convolutional block on each feature map. Convolution is capable of aggregating local information in a specific region due to its limited receptive field. In contrast to MSDA, R3FN samples feature from dense regions of the single-scale feature maps to emphasize local information. The ability of R3FN to aggregate local information is verified in the experimental section. As shown in Fig. 6, R3FN consists of three steps. First, R3FN reverses the sequence features back to 2D features according to the shape sets of the multi-scale feature maps. Second, Stand Conv (SConv) or Efficient Conv (EfConv) is applied to aggregate the local information. Finally, the aggregated feature maps are flattened into sequence features. The Stand Conv module combines a 3×3 convolutional layer, a Group Normalization layer, a GELU activation layer, and another 3×3 convolutional layer.

The SConv fuses the channel information within the local feature with low computational cost. In addition, we design the Efficient Conv module that replaces the conventional convolution in the SConv module with EConv to reduce computation. EConv divides the features equally into two parts, one part passes through the conventional convolution layer and the other part is sent to the Depthwise Separable Convolution (DwConv) [38]. Given input feature map $x_c \in R^{C \times H \times W}$, where c is the channel size of the feature map, the SConv and EfConv can be expressed as Equation (3). The computational complexity of SConv is of $O(HWC^2)$, while that of EfConv is of $O(HWC^2/2 + HWC/2)$. The features are concatenated after the convolutions and the channel size of the feature maps is preserved. We use the EfConv module for large-scale feature maps and the SConv module for small-scale feature maps.

$$\begin{aligned} SConv(x_c) &= Conv(GELU(GN(Conv(x_c)))) \\ EfConv(x_c) &= EConv(GELU(GN(EConv(x_c)))) \\ EConv(x_c) &= Concat(Conv(x_{c/2}), DwConv(x_{c/2})) \end{aligned} \tag{3}$$

C. FPQ

Each output of the decoder is put into weight-independent prediction heads for bounding box regression and class label prediction. Based on our literature review, previous works have not performed fusion operations on the intermediate decoder queries. The decoder simply uses the predicted box offsets to update the candidate boxes in the next layer. We believe that since the candidate boxes are updated iteratively layer by layer, there is a connection between these queries. However, simply concatenating or adding these queries cannot exploit the dependency information between layers and may lead to performance degradation. Therefore, we propose Fusion Previous Query (FPQ), a hierarchical query fusion mechanism based on the Geometric Sequence Sum Fusion (GSSF) and Fusion Gradient Truncation (FGT) approaches.

The structure of FPQ is shown in Fig. 7. We consider the input queries of each decoder layer as a sequence. Except for the first query, FPQ fuses the current query with all previous queries. Intuitively, the further away the previous query is, the less influence it has on the current query. Instead of

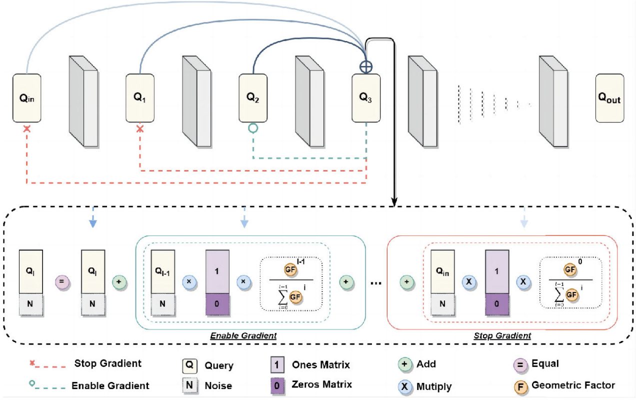


FIGURE 7. The structure of FPQ.

query fusion by simple concatenation, FPQ introduces the Geometric Sequence Sum Fusion approach. GSSF needs to set the Geometric Factor (GF) to control the fusion scales. For the l^{th} query Q_l ($l > 1$), FPQ can be achieved through Equation (4). For example, if $GF = 2$, for the 5^{th} query, the geometric sequence is [1, 2, 4, 8], and the scales for fusing the previous queries are [1/15, 2/15, 4/15, 8/15]. With $GF > 1$, the closer queries have larger fusion weights than the further queries.

$$FPQ(Q_l, GF) = \sum_{i=0}^{l-1} \frac{GF^i(GF-1)}{GF^{l-1}-1} Q_i + Q_l \tag{4}$$

The GSSF approach is effective in adjusting the fusion scale of each query, but there is a problem. Even if the GF is set to a large number, the fusion weight of the forward query is close to 0. The fusion operation causes the redundant information of the current layer to be transmitted to the previous layers through gradient backpropagation. Therefore, we use the Fusion Gradient Truncation (FGT) approach to partially truncate the negligible fusion gradient. Given a hyper-parameter called Gradient Reflow Layers (GRL), gradient backpropagation is allowed in the fusion of the current layer and the previous GRL layers. The fusion gradients of all layers further forward are truncated by the detach operation. By using both GSSF and FGT methods, FPQ effectively facilitates the integration of queries at different levels of the decoder. By adjusting GF and GRL, FPQ achieves different fusion effects. According to the following ablation experiments, GF is configured to 2, and GRL is configured to 4 in MilDetr.

IV. MILDET AND MILCLS

Unlike the common objects in large-scale visual datasets such as ImageNet, COCO [39], and Pascal VOC [40], the image data of military camouflaged targets are difficult to collect in the real world. For research purposes, we build two simulation military camouflaged target datasets called MilDet and MilCls based on Blenderproc [41].

We collect four main categories of 3D military target simulation models that are publicly available on the Internet. The main categories include armor, plane, tank, and truck, each of which has two sub-categories. We choose snowfield, grassland, and desert as the simulated backgrounds for the military camouflaged targets. By randomly selecting the simulation model and performing random scaling and rotation, we obtain simulated images of military camouflage targets in different backgrounds. In addition, we randomly change the position of the camera, the lighting of the scene, and other factors during the sampling process to enrich the diversity of MilDet. The MilDet dataset comprises images with a resolution of 960 × 540 pixels. Each image contains one or more military camouflage targets. The training set MilDetTR and the test set MilDetTE in MilDet are obtained by independent sampling without overlapping. The category information of MilDet is shown in Table 1, and more details can be found in our previous work [42].

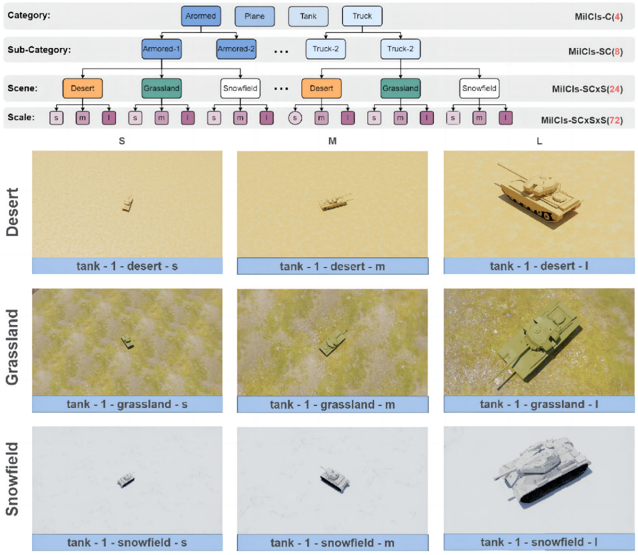


FIGURE 8. The structure and some examples of MilCls.

The MilCls dataset is made for military camouflaged target classification with the same 3D simulation models and backgrounds for MilDet. The MilCls dataset has the same image resolution as MilDet, at (960, 540). Compared with MilDet, each image in the MilCls dataset only contains one military camouflage target with random size, rotation, and illumination. The structure and some examples of MilCls are shown in Fig. 8. The training set MilClsTR and the test set MilClsTE of MilCls are constructed by two independent renderings for sampling. For MilClsTR, each 3D simulation model renders 50 images in the order of major category, subcategory, background, and size. Therefore, MilClsTR contains 3600 different training images. Similarly, MilClsTE renders 5 images for each simulation model, generating 360 test images.

V. EXPERIMENTS

We conduct experiments on the MilDet dataset. All models are trained on the MilDetTR and validated on the MilDetTE. We use the standard COCO AP metric to evaluate the performance of all methods.

TABLE 1. The details of MilDet.

Category	Armor			Plane			Tank			Truck		
Size(s/m/l)	s	m	l	s	m	l	s	m	l	s	m	l
MilDetTR	4	1103	1138	69	172	1811	27	1070	1325	17	993	1314
MilDetTE	1	113	134	9	22	183	0	103	114	2	92	133
Sum	5	1216	1272	78	194	1994	27	1173	1439	19	1085	1447

TABLE 2. The experiment environment information.

Device	Settings
CPU	AMD EPYC 7642 48-Core Processor
GPU	NVIDIA GeForce RTX 3090 * 2
Operate system	Ubuntu 20.04
CUDA	11.1
Pytorch	Pytorch 1.11.0

The experimental environment is displayed in Table 2. The ResNet-50 pre-trained on ImageNet is used as the backbone of MilDetr. The hyperparameters of both the encoder and the decoder follow Deformable DETR. We follow the training strategy of DINO and train the model for 12 epochs. We set each GPU training batch size to 2 and use AdamW as the optimizer with a learning rate of 1e-4. We use the MultiStepLR learning rate adjustment strategy with the multiplicative factor set to 0.1. Data enhancement includes random cropping and random flipping. For the comparison experiments, pre-trained ImageNet backbones are used for all models. The training strategies of the existing methods on MilDet follow their default configuration on the COCO dataset. The one-stage detectors use an input size of (640, 640). Meanwhile, the two-stage and Transformer-based detectors randomly scale the size of the input images during the training process. However, during validation and testing, the input size is fixed at (800, 1333).

A. COMPARISON WITH SOTA OBJECT DETECTORS

Table 3 compares our proposed MilDetr with SOTA object detectors. We compare the speed and accuracy of all detectors. The experimental results show that MilDetr exhibits SOTA performance on MilDet with 95.6 AP. In the experiments, the top-performing detector among the SOTA models is the two-stage detector, Cascade-RCNN, which achieves 94.6 AP. However, Cascade-RCNN has a slower detection speed and large parameters. YOLO detectors have lower Params and higher GFLOPs. YOLOv5-s and YOLOv7-t have poor detection performance. YOLOv6-s and YOLOv8-s have comparable performance to the two-stage detectors. Furthermore, there are significant performance differences among the Transformer-based detectors. DETR had the lowest detection performance at 78.7 AP, followed by Conditional-DETR and DAB-DETR. Deformable DETR with MSDA achieves 90.4 AP. DINO, which applies the CDN training approach, achieves 92.6 AP under 12 training epochs. Deformable DETR and DINO show relatively good performance but are still not as good as Cascade-RCNN.

Compared to YOLOv5-s, YOLOv6-s, YOLOv7-t, and YOLOv8-s, MilDetr improves accuracy significantly by 19.0%, 5.7%, 28.7%, and 1.9% AP respectively. Compared to Faster-RCNN and Cascade-RCNN, MilDetr achieves an improvement in accuracy of 3.4% AP and 1.1% AP respectively. MilDetr achieves an improvement in accuracy of 21.5%, 8.3%, 9.6%, 5.7%, and 3.2% AP compared to DETR, Conditional-DETR, DAB-DETR, Deformable DETR, and DINO, respectively. Compared to YOLO detectors, MilDetr shows a significant improvement in detection performance and slower detection speed. Although MilDetr outperforms YOLOv8-s in overall accuracy, its accuracy in detecting small objects is inferior to that of YOLOv8-s. Compared to the two-stage detector and the Transformer-based detector, MilDetr demonstrates improved detection performance and similar detection speed.

B. ABLATION EXPERIMENTS

We validate the effectiveness of our improved methods on MilDet. The results of the various methods with optimal hyper-parameter settings are available in Table 4.

We first verify the effectiveness of MDSA and CDN. Both MDSA and CDN significantly improve the performance of model detection when used individually. However, the method combining both MDSA and CDN performs worse than Deformable DETR and DINO. We suggest that the deformed receptive fields produced by MDSA may partially overlap with the receptive fields of the negative samples produced by CDN, causing problems for model training.

We verify the effectiveness of R3FN in improving detection performance through ablation experiments. We use R3FN to reintroduce local information after MDSA in the encoder of MilDetr. By using R3FN, MilDetr achieves 93.1 AP. We then evaluate the effectiveness of FPQ. We use FPQ in the decoder of MilDetr to fuse multi-stage query features. By using FPQ, the accuracy increases to 95.6 AP without extra training parameters. The proposed MilDetr achieves the best results, suggesting that our proposed modules effectively mitigate the aforementioned problem and improve the performance of military camouflaged target detection.

1) R3FN

To solve the sparse receptive field problem, we propose R3FN and present the EfConv module to reduce the number of model parameters. To explore the structure of R3FN, we design ablation experiments with different feature layers using different convolutional blocks, and the experimental results are shown in Table 5. In the table, F1-F4 denotes feature maps of different sizes after being reversed from 1D features. F1 is the feature map with the largest size and F4

TABLE 3. Comparisons results with SOTA object detectors on MilDet.

Method	Input size	Params(M)	Epochs	GFLOPs	$FPS_{bs=1}$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLOv5-s	(640, 640)	7.05	300	7.94	110.9	80.3	97.5	94.4	39.2	81.5	83.2
YOLOv6-s		18.83	300	24.199	103.0	90.4	98.5	96.4	43.0	92.2	92.7
YOLOv7-t		6.04	300	6.59	**114.8**	74.3	96.3	88.8	31.9	75.8	77.2
YOLOv8-s		11.16	300	14.27	110.8	93.8	98.8	97.9	**56.8**	94.8	96.1
Faster-RCNN	(800, 1333)	41.42	12	216.31	88.2	92.5	99.0	97.3	37.2	94.1	93.5
Cascade-RCNN		69.21	12	244.11	88.3	94.6	98.7	97.8	41.2	95.7	96.7
DETR	(800, 1333)	41.61	150	96.17	86.9	78.7	97.1	92.2	44.2	80.0	82.1
Conditional-DETR		43.50	50	100.13	88.3	81.6	97.9	93.4	20.8	83.4	94.8
DAB-DETR		43.76	12	101.64	87.3	87.2	98.4	95.2	42.4	87.9	90.8
Deformable DETR		40.82	12	204.91	88.9	90.4	99.2	97.0	47.8	91.6	92.8
DINO		47.60	12	287.32	87.2	92.6	98.7	96.0	45.9	93.4	94.9
Mildetr		62.25	12	230.74	88.5	**95.6**	**99.4**	**98.1**	52.9	**96.2**	**97.6**

TABLE 4. Ablation results of various methods on MilDetTE.

Method	MSDA	R3FN	FPQ	CDN	Params(M)	$FPS_{bs=1}$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
DETR					41.61	86.9	78.7	97.1	92.2	44.2	80.0	82.1
Deformable DETR	✓				40.82	88.9	90.4	99.2	97.0	47.8	91.6	92.8
DINO				✓	47.60	87.2	92.6	98.7	96.0	45.9	93.4	94.9
MilDetr (without R3FN and FPQ)	✓			✓	44.12	88.7	90.0	96.6	94.2	45.6	91.1	92.3
MilDetr (without FPQ)	✓	✓		✓	62.25	88.5	93.1	98.4	96.7	49.3	94.3	94.9
MilDetr	✓	✓	✓	✓	62.25	88.5	**95.6**	**99.4**	**98.1**	**52.9**	**96.2**	**97.6**

is the feature map with the smallest size. e indicates the use of the EfConv module and s indicates the use of the SConv module.

From Table 5, it can be seen that the performance of the network model after using the R3FN strategy has improved considerably compared to the Baseline experiments using FFN. In S1, the model has a 2.6% performance improvement after replacing FFN with R3FN. To reduce the number of parameters, S2-S5 sequentially replaces the SConv with EfConv. The results in Table 5 show that EfConv can significantly reduce the number of parameters. S5 replaces all SConv modules with the EfConv modules. Compared to S1, S5 has a 20.37% reduction in parameter amount and a 0.7% reduction in detection performance.

In S3, only two SConv modules are replaced with EfConv modules in R3FN, and the model achieves 93.1 AP. The comparison of the Precision-Recall curves of baseline and S3 is shown in Fig. 9. By using R3FN, the blue and purple areas in the Precision-Recall curve decreased significantly, indicating that the model has better positioning ability. Especially, the area of the purple region decreased a lot. It proves that R3FN is capable of avoiding misdetection of the background. As shown in Fig. 10, MilDetr focuses more on the neighborhood of the targets, confirming R3FN's ability to aggregate local information. The experimental results show the effectiveness of the R3FN, and subsequent ablation experiments are conducted based on the configuration of S3.

2) FPQ

For better feature fusion, we propose FPQ, which includes GSSF and FGT. There are two hyper-parameters in FPQ: Geometric Factor (GF) and Gradient Reflow Layers (GRL). By adjusting these two hyper-parameters, FPQ achieves different fusion effects. We conduct comparison experiments on these two hyper-parameters with different settings.

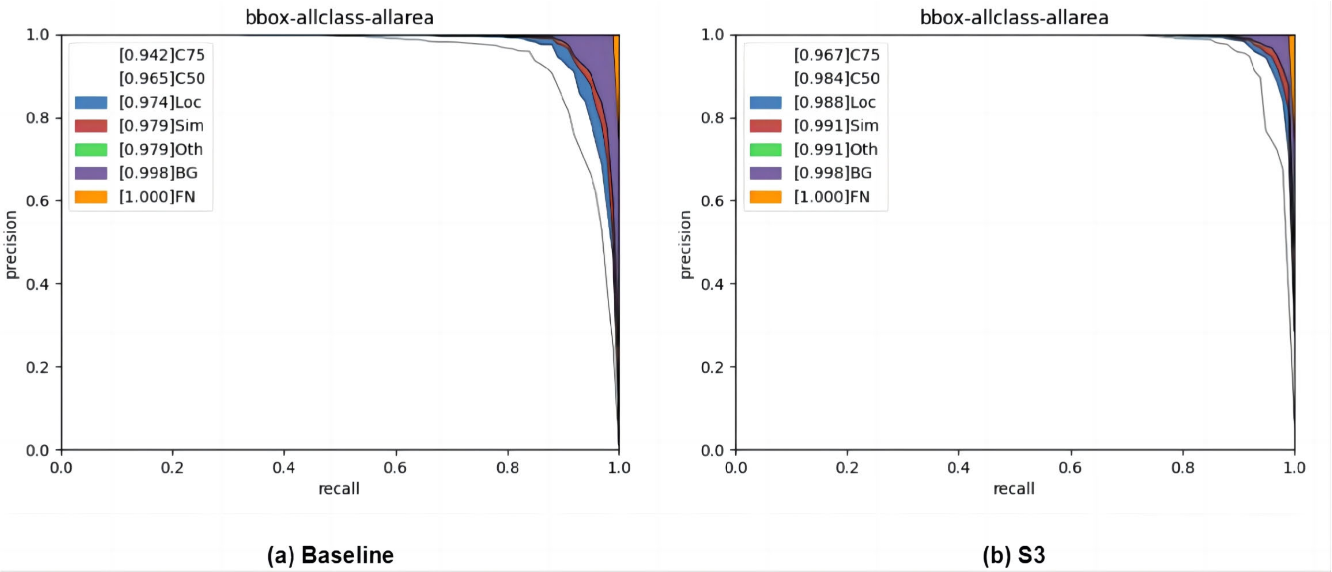
Firstly, We select 0, 1, 2, 3, and 4 as GF for comparative experiments, and the experimental results are shown in Table 6. If $GF = 0$, the decoder queries are not fused. When $GF = 1$, the decoder queries are fused proportionally. When $GF > 1$, queries are fused using the sum of the geometric sequence. When GF=1, the performance of the model decreases by 0.4% compared to the model without query fusion. The result indicates that the simple query fusion brings too much redundant information, leading to a performance flop. The model's performance is improved by 2.0% AP when using GSSF with $GF = 4$ compared to model without query fusion. Other models that have applied GSSF also perform better than the model without query fusion.

When GF is set to 2, 3, and 4, we explore the setting of GRL. MilDetr has 6 decoder layers, so the maximum number of gradient propagation layers is 5, which means $GRL \in \{1, 2, 3, 4, 5\}$. When the FGT strategy is not used, GRL is set to 5, which means that gradient propagation is performed between all queries. The experimental results are shown in Table 7. According to the results, the complex gradient propagation leads to a decrease in model performance. Truncating the partial gradient propagation by reducing the GRL can significantly improve performance. When GRL=2, the model performance reaches 95.5 AP, which improved by 0.6% compared to the model without gradient truncation. When GF=3, reducing the number of gradient propagation layers to 3, the model achieves 95.6 AP. Therefore, when the number of gradient propagation layers is set to 3, the model achieves better performance.

From the experimental results, it can be seen that choosing an appropriate fusion method for queries in decoder can preserve important information, and truncating redundant

TABLE 5. The ablation results on R3FN.

Setting	F1	F2	F3	F4	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params(M)
Baseline	FFN				90.0	96.6	94.2	45.6	91.1	92.3	216.96	44.12
S1	*s*	*s*	*s*	*s*	92.6	97.2	95.8	44.5	93.6	**95.3**	304.73	69.30
S2	*e*	*s*	*s*	*s*	92.2	97.7	96.5	**52.2**	92.9	94.8	245.54	65.78
S3	*e*	*e*	*s*	*s*	**93.1**	**98.4**	**96.7**	49.3	**94.3**	94.9	230.74	62.25
S4	*e*	*e*	*e*	*s*	91.7	97.3	95.3	43.6	93.0	93.5	227.04	58.73
S5	*e*	*e*	*e*	*e*	91.9	98.1	96.0	47.6	93.5	93.3	226.08	55.20



(a) Baseline

(b) S3

FIGURE 9. The precision-recall curves of baseline and MilDetr applying R3FN with settings in S3.

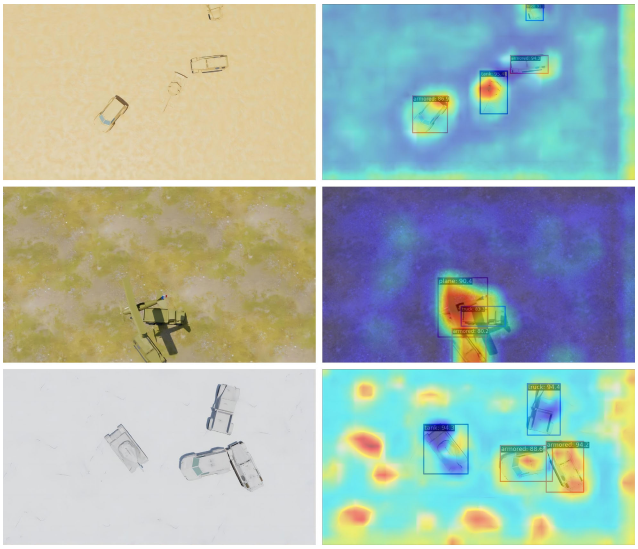


FIGURE 10. Input images (col 1) and visualization of the attention maps of MilDetr (col 2).

gradient propagation can also simplify the model training process.

TABLE 6. The ablation results on FPQ with different *GF* settings.

GF	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0	93.1	98.4	96.7	49.3	94.3	94.9
1	92.7	98.2	96.6	43.3	93.9	94.2
2	94.9	**99.2**	97.3	**52.8**	95.5	96.6
3	94.6	98.8	97.2	52.3	95.3	96.6
4	**95.1**	99.0	**97.6**	51.0	**95.6**	**97.2**

TABLE 7. The ablation results on FPQ with different *GRL* settings.

GF	GRL	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
2	1	95.4	99.4	98.0	45.1	96.1	96.7
	2	95.5	99.4	98.0	59.3	96.1	97.5
	3	94.6	99.4	97.6	51.8	95.3	96.5
	4	94.8	99.3	97.4	47.4	95.5	97
	5	94.9	99.2	97.3	52.8	95.5	96.5
3	1	95.4	99.3	97.8	57.1	95.7	97.6
	2	95.4	99.3	97.7	54.3	95.9	97.6
	3	**95.6**	99.4	98.1	52.9	96.2	97.6
	4	94.9	99.2	97.5	49.2	95.5	97.4
	5	94.6	98.8	97.2	52.3	95.3	96.6
4	1	95.3	99.3	98.0	52.6	95.7	97.7
	2	95.6	99.3	97.6	57.3	96.4	97.2
	3	95.3	99.5	98.0	54.3	96.2	96.7
	4	95.2	99.3	98.1	51.5	95.8	97.0
	5	95.1	99.0	97.6	51.0	95.6	97.2

3) PRE-TRAINING ANALYSIS

Pre-training refers to training a model on a large dataset and then using its ability for other tasks. The main advantage of pre-training is that the model can learn rich common features and improve generalization ability. We further perform pre-training experiments on the DETR series with the different pre-training settings: training from scratch, using weights pre-trained on MilCls, and using weights

TABLE 8. The experimental results of various models with different pre-training settings.

Method	Pretrained	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Conditional DETR	Scripts	50	37.7	69.9	36.9	5.5	41.3	40.9
	MilCls	50	73.4	95.5	59.1	8.6	75.8	75.4
	ImageNet	50	**81.6**	**97.9**	**93.4**	**44.2**	**80.0**	**82.1**
DAB-DETR	Scripts	50	35.3	65.8	32.6	13.1	38.3	36.3
	MilCls	50	74.9	96.6	88.1	18.5	76.8	77.6
	ImageNet	50	**87.2**	**97.9**	**93.4**	**20.8**	**83.4**	**94.8**
Deformable DETR	Scripts	50	85.6	98.1	95.5	43.9	87.1	88.0
	MilCls	50	85.2	98.4	94.6	24.5	86.9	87.7
	ImageNet	50	**90.4**	**99.2**	**97.0**	**47.8**	**91.6**	**92.8**
DINO	Scripts	12	89.7	97.7	95.1	46.2	90.8	92.0
	MilCls	12	90.4	98.2	94.8	**54.6**	91.4	92.4
	ImageNet	12	**92.6**	**98.7**	**96.0**	45.9	**93.4**	**94.9**

TABLE 9. The comparison of DINO and MilDetr with different pre-trianing strategies.

	Pretrained	Epochs	AP	AP50	AP75	APS	APM	APL
DINO	MilCls	12	90.4	98.2	94.8	54.6	91.4	92.4
	ImageNet	12	92.6	98.7	96.0	45.9	93.4	94.9
	ImageNet+MilCls	12	**93.5**	**99.0**	**97.2**	**54.7**	**94.1**	**95.8**
MilDetr	ImageNet	12	95.6	99.4	98.1	52.9	96.2	97.6
	ImageNet+MilCls	12	**96.4**	**99.4**	**98.2**	**63.8**	**97.1**	**97.9**

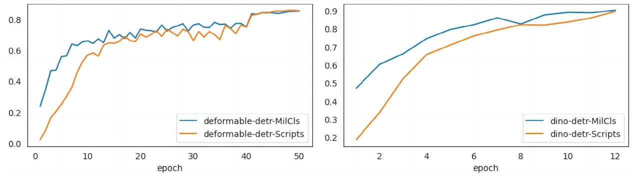


FIGURE 11. The AP curves of Deformable DETR and DINO.

pre-trained on ImageNet. DETR is not included due to its slow convergence speed. The experimental results for the models are shown in Table 8.

The Conditional DETR and DAB-DETR algorithms using only single-scale features without pre-training have extremely poor performance, with only 37.7 AP and 35.5 AP respectively. Deformable DETR and DINO using multi-scale feature layers achieve 85.6 AP and 89.7 AP, respectively.

By using pre-trained weights on the MilCls dataset, Conditional DETR has about 36% improvement and DAB-DETR has 39% improvement, demonstrating the effectiveness of pre-training. For network models using multi-scale features, DINO has only 0.7% improvement, and Deformable DETR decreases by 0.4%. Fig. 11 shows the AP curves of the Deformable DETR and DINO without pre-training and pre-training with MilCls. It can be seen that although MilCls pre-trained weights do not bring significant performance improvements to Deformable DETR and DINO, the convergence speed of both models is significantly accelerated. The prediction performance of MilCls pre-trained wights cannot be comparable to that of the ImageNet pre-trained wights. The reason for this may be that MilCls has only 3600 training images, which is much fewer than in ImageNet.

We select the best-performing DINO in the Transformer-based object detectors and the proposed MilDetr for further experiments. The backbone of both models is the ImageNet pre-trained ResNet50 and then finetuned on MilCls for 25 epochs with the weights of the first three blocks frozen. The experimental results in Table 9 show that both models have improved. Specifically, DINO has a 0.9 AP improvement while MilDetr has a 0.8 AP improvement.

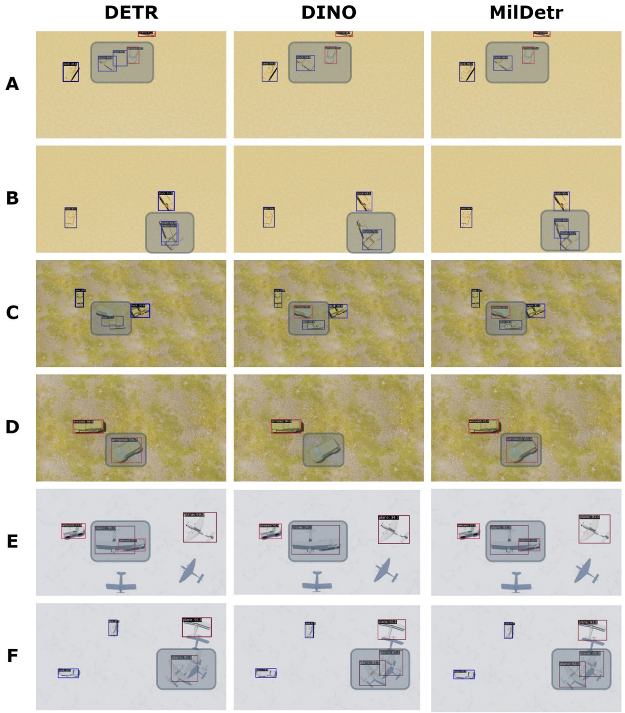


FIGURE 12. The visualization detection results of DETR, DINO and MilDetr.

Experiments demonstrate the importance of pre-training for downstream tasks in large models. In future work, we will expand the MilCls dataset to serve as a large-scale pre-training dataset for MCTD.

C. VISUALIZATION ANALYSIS

We visualize the military camouflaged target detection results of DETR, DINO, and MilDetr in the MilDetTE, and

TABLE 10. Paired samples t-test on YOLOv8-s, Cascade-RCNN, DINO and Mildetr.

Methods	No.	AP	$\bar{x}_{diff}$	statistic	p-value
MilDetr	1	95.6	–	–	–
	2	95.6			
	3	95.2			
	4	95.5			
	5	95.6			
YOLOv8-s	1	93.8	1.86	18.43	5.09e-5
	2	93.7			
	3	93.7			
	4	93.6			
	5	93.4			
Cascade-RCNN	1	94.6	1.06	9.32	7.34e-4
	2	94.6			
	3	94.0			
	4	94.4			
	5	94.6			
DINO	1	91.9	3.14	16.08	8.74e-5
	2	92.2			
	3	92.5			
	4	92.6			
	5	92.6			

five different examples are shown in Fig. 12 (a-f). DETR encounters the background misdetection problem in (a) and the omission problem in (b), (c), and (f). DINO has omission problems in (b), (d), and (e). At the same time, MilDet does not encounter any issues in these instances, indicating that MilDetr's military camouflaged target detection performance is significantly better than DETR and DINO.

D. STATISTICAL SIGNIFICANCE ANALYSIS

To determine the effectiveness of MilDetr, we perform a paired samples t-test at significance level $\alpha = 0.05$ using AP metric. We choose YOLOv8-s, Cascade-RCNN and DINO as the best representatives of each type of detector and perform five replicated experiments on MilDet. Given the five sets of accuracies for each model, we calculate the p-value between each model and MilDetr independently. Table 10 shows the results of the experiments. Since all p-values are less than the significance level $\alpha = 0.05$, we validate that MilDetr outperforms other object detectors.

VI. CONCLUSION

In this paper, two simulation datasets called MilDet and MilCls are constructed. Through experiments on MilDet, we find that Transformer-based object detectors, which perform well in detecting general objects do not perform as well in MCTD. We propose the end-to-end Military Detection Transformer (MilDetr) by integrating Multi-scale Deformable Attention (MSDA), R3FN, and FPQ into DETR. First, We design the Reverse Features Feed Forward Neural Network (R3FN) in the encoder of MilDetr for local information aggregation. Furthermore, we use the Fusion Previous Query (FPQ) module in the decoder of MilDetr for multi-stage query feature fusion.

The ablation experiments demonstrate the effectiveness of R3FN and FPQ. The comparative experiments with the SOTA object detectors demonstrate that MilDetr can effectively detect military camouflage targets. In addition, we demonstrate the importance of pre-training for downstream tasks, where pre-training on ImageNet and fine-tuning on MilCls can bring significant performance and convergence speed improvements to the object detection model and the convergence speed of models. With the pre-trained weight on ImageNet and MilCls, MilDetr achieves the best performance with 96.4 AP on MilDetTE.

The experimental results illustrate that MilDetr achieves state-of-the-art (SOTA) performance on the MilDet. However, compared with YOLO detectors, MilDetr has a slow detection speed and large parameters. Our future work will focus on expanding the dataset for MCTD and improving the detection speed of MilDetr. We believe that model compression is a promising direction to pursue. We hope that our initial efforts can accelerate the development of end-to-end object detectors on MCTD.

REFERENCES

[1] M. Shah, O. Javed, and K. Shafique, "Automated visual surveillance in realistic scenarios," *IEEE Multimedia Mag.*, vol. 14, no. 1, pp. 30–39, Jan. 2007.

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015.

[3] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[4] L. Talas, R. J. Baddeley, and I. C. Cuthill, "Cultural evolution of military camouflage," *Philos. Trans. Roy. Soc. B, Biol. Sci.*, vol. 372, no. 1724, May 2017, Art. no. 20160351.

[5] C. J. Lin, Y. T. Prasetyo, N. D. Siswanto, and B. C. Jiang, "Optimization of color design for military camouflage in CIELAB color space," *Color Res. Appl.*, vol. 44, no. 3, pp. 367–380, Jun. 2019.

[6] T. T. Brunyé, M. D. Eddy, M. S. Cain, L. B. Hepfinger, and K. Rock, "Masked priming for the comparative evaluation of camouflage conspicuity," *Appl. Ergonom.*, vol. 62, pp. 259–267, Jul. 2017.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448.

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

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

[11] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 21–37.

[13] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 6517–6525.

[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

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

[18] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, Jun. 2023, pp. 7464–7475.

[20] 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.

[21] D. Meng, X. Chen, Z. Fan, Z. G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional DETR for fast training convergence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 3631–3640.

[22] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, and L. Zhang, "DAB-DETR: Dynamic anchor boxes are better queries for DETR," 2022, *arXiv:2201.12329*.

[23] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.

[24] 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/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 13609–13617.

[25] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved denoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.

[26] X. Song, "Overview of military target recognition algorithms based on deep learning," *Sci. Technol. Eng.*, vol. 22, no. 22, pp. 9466–9475, 2022.

[27] Y. Qin and B. Li, "Effective infrared small target detection utilizing a novel local contrast method," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1890–1894, Dec. 2016.

[28] M. T. Eismann, C. R. Schwartz, J. N. Cederquist, J. A. Hackwell, and R. J. Huppi, "Comparison of infrared imaging hyperspectral sensors for military target detection applications," in *Imaging Spectrometry II*, vol. 2819. Bellingham, WA, USA: SPIE, Nov. 1996, pp. 91–101.

[29] A. Tankus and Y. Yeshurun, "Detection of regions of interest and camouflage breaking by direct convexity estimation," in *Proc. IEEE Workshop Vis. Surveill.*, Bombay, India, Jan. 1998, pp. 42–48.

[30] X. Zhang, C. Zhu, S. Wang, Y. Liu, and M. Ye, "A Bayesian approach to camouflaged moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 9, pp. 2001–2013, Sep. 2017.

[31] Y. Beiderman, M. Teicher, J. Garcia, V. Mico, and Z. Zalevsky, "Optical technique for classification, recognition and identification of obscured objects," *Opt. Commun.*, vol. 283, no. 21, pp. 4274–4282, Nov. 2010.

[32] S. Galun and B. Basri, "Texture segmentation by multiscale aggregation of filter responses and shape elements," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, vol. 1, Oct. 2003, pp. 716–723, doi: 10.1109/ICCV.2003.1238418.

[33] H. Guo, Y. Dou, T. Tian, J. Zhou, and S. Yu, "A robust foreground segmentation method by temporal averaging multiple video frames," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Shanghai, China, Jul. 2008, pp. 878–882, doi: 10.1109/ICALIP.2008.4590132.

[34] B. Yu, "Improved YOLOv3 algorithm and its application in military target detection," *Acta Armamentarii*, vol. 43, no. 2, p. 345, 2022.

[35] X. Deng, T. Cao, and Z. Fang, "Research on improved RetinaNet camouflaged person detection method," *Comput. Eng. Appl.*, vol. 57, no. 5, pp. 190–196, 2021.

[36] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[38] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1800–1807.

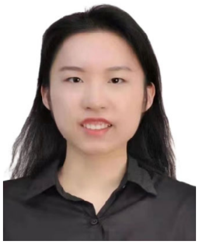
[39] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, 2014, pp. 740–755.

[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, Jun. 2010.

[41] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, "BlenderProc," 2019, *arXiv:1911.01911*.

[42] B. Li, E. Zhu, R. Zhou, and H. Cheng, "MilInst: Enhanced instance segmentation framework for military camouflaged targets using sparse instance activation," *IEEE Access*, vol. 11, pp. 106387–106396, 2023.

BING LI received the B.S., M.S., and Ph.D. degrees from the Mechanical Engineering College, Shijiazhuang, China, in 2003, 2006, and 2010, respectively. He is currently an Associate Professor with the Department of Electrical and Information Engineering, Shantou University. His research interests include intelligent computing, computer vision, deep learning, and signal and image processing.



RONGQIAN ZHOU was born in Shaoguan, China, in 1999. She is currently pursuing the master's degree in electrical information with Shantou University, Shantou, China. Her main research interests include computer vision and natural language processing.

LU YANG, photograph and biography not available at the time of publication.

QIWEN WANG, photograph and biography not available at the time of publication.

HUANG CHEN was born in Fuzhou, China, in 1997. He received the master's degree in electronic information (major) from Shantou University, Shantou, China. His main research interests include deep learning, object detection, and instance segmentation.

• • •