CamoFormer: Masked Separable Attention for Camouflaged Object Detection

Bowen Yin, Xuying Zhang, Deng-Ping Fan, *Senior Member, IEEE*, Shaohui Jiao, Ming-Ming Cheng, *Senior Member, IEEE*, Luc Van Gool, Qibin Hou

Abstract—How to identify and segment camouflaged objects from the background is challenging. Inspired by the multi-head self-attention in Transformers, we present a simple masked separable attention (MSA) for camouflaged object detection. We first separate the multi-head self-attention into three parts, which are responsible for distinguishing the camouflaged objects from the background using different mask strategies. Furthermore, we propose to capture high-resolution semantic representations progressively based on a simple top-down decoder with the proposed MSA to attain precise segmentation results. These structures plus a backbone encoder form a new model, dubbed CamoFormer. Extensive experiments show that CamoFormer achieves new state-of-the-art performance on three widely-used camouflaged object detection benchmarks. To better evaluate the performance of the proposed CamoFormer around the border regions, we propose to use two new metrics, *i.e.*, BR-M and BR-F. There are on average ∼**5%** relative improvements over previous methods in terms of S-measure and weighted F-measure. Our code is available at https://github.com/HVision-NKU/CamoFormer.

Index Terms—camouflaged object detection, self-attention, masked separable attention, top-down decoder

1 Introduction

Foreground and background (FG/BG) segmentation techniques [5], [22], [39] play a crucial role in computer vision, aiming to accurately distinguish and separate the main subject (foreground) from the surrounding environment (background), contributing to advancements in object recognition [3], scene understanding [4], [33], *etc*. Camouflaged object detection (COD) is a new challenging FG/BG segmentation task [13] that has been popular in recent years [13], [14], [16], [26], [51]. Biological studies have shown that the human visual perceptual system can be easily deceived [57] by various camouflage strategies in that camouflaged objects are highly similar to their surroundings or extremely small in size. The high similarity between the camouflaged objects and their surroundings makes COD [14] more challenging than traditional object detection tasks [38], [50]. It has been proven beneficial to applications in different fields of art (e.g., photo-realistic blending [18] and recreational art [8]) and medical diagnosis (e.g., polyp segmentation [16]).

There is an increasing number of works using sophisticated deep learning techniques [51], [52], [79] to solve this task, especially after a large-scale dataset was proposed [13]. However, even the state-of-the-art methods still struggle to segment camouflaged targets with fine shapes for some complex scenes because of the characteristics of camouflaged objects as mentioned above. Some examples are shown in Fig. 1. A promising way to better identify the camouflaged objects from similar surroundings is to separately encode the foreground and background cues and highlight the contrasting information instead of coping with the foreground and background cues indiscriminately.

As a successful attempt, PFNet [51] reveals that false

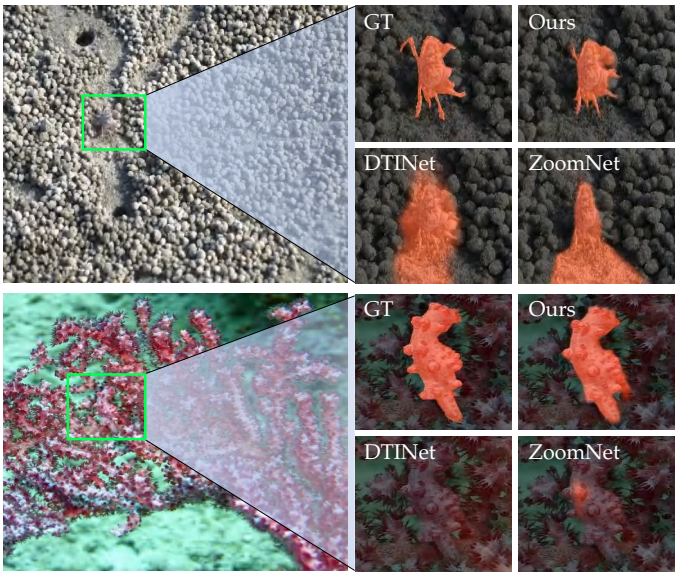


Fig. 1. Visual comparisons between our CamoFormer and recent state-of-the-art methods (e.g., DTINet [44] and ZoomNet [52]) for camouflaged object detection. Recent state-of-the-art methods still struggle to capture the camouflaged objects and distinguish similar backgrounds around them, while our CamoFormer can capture and segment the targets more accurately. Best viewed in color.

- *This research was supported by NSFC (NO. 62225604, No. 62276145), the Fundamental Research Funds for the Central Universities (Nankai University, 070-63223049). Computations were supported by the Supercomputing Center of Nankai University (NKSC).*
- *B. Yin, X. Zhang, D.P. Fan, M.M. Cheng, and Q. Hou are with VCIP, School of Computer Science, Nankai University, Tianjin, China (bowenyin@mail.nankai.edu.cn, zhangxuying1004@gmail.com). D.P. Fan, M.M. Cheng, and Q. Hou are also with Nankai International Advanced Research Institute (Shenzhen Futian).*
- *S. Jiao is with Bytedance Inc.*
- *L. V. Gool is with CVL, ETH Zurich, Switzerland.*
- *Q. Hou is the corresponding author (andrewhoux@gmail.com).*
- *First two authors contributed equally.*

Manuscript received March 1, 2022; revised August 26, 2022.

positive and false negative predictions often naturally occur in the segmentation results. To alleviate this, a distraction mining strategy is developed in [51] to separately process the features of the target and background so as to remove these false predictions. Despite its good performance, PFNet only focuses on local features for point-level refinement but neglects the importance of building interactions between the foreground and background features.

In this paper, we present *Masked Separable Attention (MSA)*, which considers the way of encoding camouflaged objects and background features from a new perspective. Our MSA is built upon the multi-head self-attention mechanism but unlike traditional methods that utilize multiple attention heads simply for enhancing the feature representations, we propose to leverage different attention heads to calculate pixel correlations for different regions. To be specific, we split the self-attention heads into three groups. We first use two groups of heads to compute pixel correlations of the foreground and background regions independently. Our goal is to use the attention scores built within the predicted foreground generated by a prediction head to index camouflaged objects from the full-value representations and similarly for the background. Besides, we preserve a group of normal attention heads for computing pixel correlations of the full map, which can help distinguish the camouflaged objects from a global view. Thus, three groups of heads are complementary.

Given the proposed MSA, we apply it to an encoder-decoder architecture [36], [37], [40] to progressively refine the segmentation map as illustrated in Fig. 2. At each feature level of the decoder, a segmentation map is predicted and sent to an MSA block to improve the prediction quality. This progressive refinement process enables us to attain high-quality camouflaged object predictions as the feature resolution increases. As shown in Fig. 1, our CamoFormer can more accurately identify the camouflaged objects and generate segmentation maps with finer borders than other cutting-edge methods.

To validate the effectiveness of CamoFormer, we conduct extensive experiments on three popular COD benchmarks (NC4K [47], COD10K [14], and CAMO [34]). On all these benchmarks, our CamoFormer achieves new state-of-the-art records compared to recent cutting-edge methods. In particular, our method achieves 0.793 weighted F-measure and 0.022 MAE, while the corresponding results for the second-best model FDNet [79] are 0.731 and 0.030 on the COD10K-test set. Furthermore, we carry out comprehensive visualization experiments whose results also show the superiority of our CamoFormer over existing COD methods.

In addition to traditional measurements, regarding that camouflaged objects are quite similar to their surrounding regions and their scales are often small, we also measure the segmentation quality around the border regions of the camouflaged objects. In particular, we propose to compute the weighted F-measure and mean absolute error scores just around the object borders, resulting in two new metrics, namely BR-M and BR-F. Experiments show that on these two evaluations of border regions, our CamoFormer performs even better than other state-of-the-art methods.

Our main contributions can be summarized as follows:

- We present masked separable attention (MSA), a novel method that uses different self-attention heads to compute visual similarity for different regions and meanwhile explicitly model the global dependencies between foreground and background.
- We present a new network architecture, termed CamoFormer, where a top-down path is built to exploit the full potential of our MSA. Experimental results show that our method achieves better performance than previous works.
- We propose two new simple metrics to evaluate the performance of camouflaged object detection models on border regions and show that our method performs better at processing boundary areas.

2 Related Work

2.1 Camouflaged Object Detection

Traditional COD methods [1], [17], [20], [21], [31], [77] extract various hand-crafted features between the camouflaged objects and backgrounds to segment the camouflaged targets. These methods can deal with simple scenes but show drastic accuracy degradation in complex conditions. Developing this field via deep learning methods [6], [15], [45], [46], [76] has become the current trend.

Recently, the mainstream in COD is CNN-based approaches [14], [27], [28], [51], [52], [74], [79], which can be categorized into three strategies: i) Multi-scale feature aggregation: CubeNet [82] accompanies attention fusion and X-shaped connection to integrate features from multiple layers sufficiently. ZoomNet [52] processes the input images at three scales and unifies the scale-specific appearance features at different scales. ii) Multi-stage strategy: Due to the concealment of camouflaged objects, SINet [14] proposed first to locate and then distinguish them for better performance. PreyNet [74] mimics the process of predation and splits the detection process of camouflaged targets into initial detection and predator learning. SINetV2 [13] adopts surrounding connection decoder and group-reversal attention to improve the performance. SegMaR [28], a multi-stage training and inference framework, locates the target and magnifies the object regions to detect camouflaged objects progressively. iii) Joint training strategy: UJSC [35] leverages the contradictory information to enhance the detection ability for both salient object detection and camouflaged object detection. SLSR [47] combines camouflaged ranking and camouflaged object detection to construct the joint-training framework. However, these methods are still powerless for camouflaged objects with complex shapes.

Our work is also related to the PFNet [51] work. However, PFNet simply splits the feature maps via a distraction mining strategy, and the features of the target and background are separately processed to remove the false predictions. Different from this method, our CamoFormer takes advantage of self-attention and computes the visual similarity for different regions using different self-attention heads to index the targets from the whole feature maps.

2.2 Transformers in Computer Vision

Compared with conventional convolutional neural networks [23], [25], [55], [56], [60], Transformers can effi-

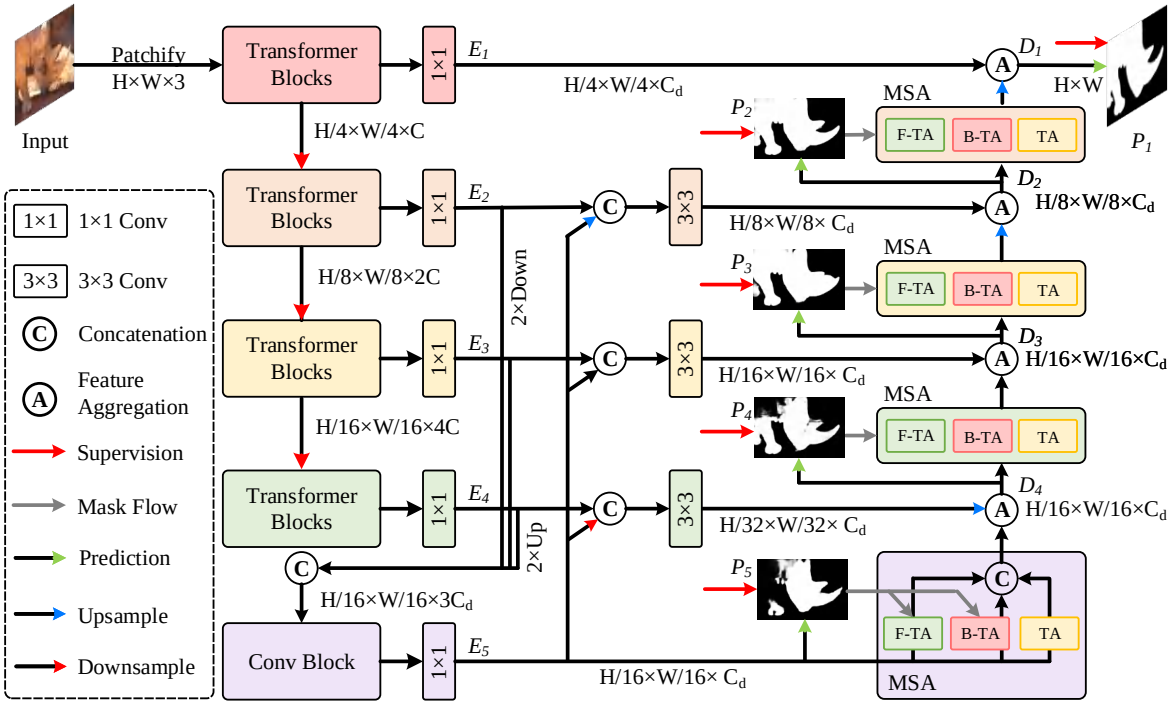


Fig. 2. Overall architecture of our CamoFormer model. First, a pretrained Transformer-based backbone is utilized to extract multi-scale features of the input image. Then, the features from the last three stages are aggregated to generate the coarse predictions. Next, the progressive refinement decoder equipped with masked separable attention (MSA) is applied to gradually polish the prediction results. F-TA, B-TA, and TA heads separately calculate the attention scores in the predicted foreground, background, and the whole image and MSA uses them to identify the camouflaged objects better. All the generated predictions are supervised by the ground truth.

ciently encode global contextual information and hence have been widely used in a variety of visual tasks, including image classification [10], [61], [64], [70], semantic segmentation [30], [66], [69], [78], object detection [2], super-resolution [80], and salient object detection [19], [41], [81].

Transformer-based models are also becoming a new trend in COD. UGTR [68] explicitly utilizes the probabilistic representational model to learn the uncertainties of the camouflaged object under the Transformer framework. DTINet [44] designs a dual-task interactive Transformer to segment both the camouflaged objects and their detailed borders. TPRNet [75] proposes a transformer-induced progressive refinement network that utilizes the semantic information from high-level features to guide the detection of camouflaged targets. In addition, under the Transformer-based framework, SLTNet [7] exploits short-term dynamics and long-term temporal consistency to capture dynamic camouflaged objects in videos.

Our CamoFormer is also built upon the popular Transformer framework. Not focusing on a novel architecture design, we aim to investigate more efficient ways to utilize self-attention for COD and receive better performance than other methods. We assign different functionalities to different attention heads to process the foreground and background regions separately, which makes our work quite different from other Transformer-based COD methods.

3 Proposed CamoFormer

3.1 Overall Architecture

Similar to most previous works [14], [16], [29], [52], [79], we adopt an encoder-decoder architecture to build our CamoFormer, which is shown in Fig. 2.

Encoder. By default, we adopt PVTv2 [65] as our encoder, as vision transformers have shown great performance in binary segmentation tasks [41], [73]. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, we feed it into the encoder to generate multi-scale feature maps from the four stages, which are denoted as $\{E_i\}_{i=1}^{4}$. Consequently, E_1 is with spatial size $\frac{H}{4}\times\frac{W}{4}$ and E_4 is with spatial size $\frac{H}{32}\times\frac{W}{32}$. Then, we aggregate the features from the last three stages of the encoder and send them to a convolutional block, yielding representations E_5 with higher-level semantics.

Decoder. The decoder is built upon the encoder. The multi-level semantic features $\{E_i\}_{i=1}^{5}$ from the encoder are fed into the decoder. To achieve a better trade-off between efficiency and performance, we first connect a 1×1 convolution with $C_d = 128$ channels to the feature maps at each level. As shown in Fig. 2, we adopt a progressive way to refine the features from the top of the encoder. At each feature level, masked separable attention (MSA) is used for a better distinguishment of the camouflaged objects and the background. In the initial level of progressive fusion, the aggregated feature D_4 can be written as:

$$D_4 = \mathrm{MSA}(E_5)\cdot\mathcal{F}_{\mathrm{up}}(E_4) + \mathcal{F}_{\mathrm{up}}(E_4), \tag{1}$$

where $\mathcal{F}_{\mathrm{up}}(\cdot)$ is a bilinear upsampling operation for shape matching. The aggregated features $\{D_i\}_{i=1}^{3}$ in the following levels can be defined as:

$$D_i = \mathcal{F}_{\mathrm{up}}(\mathrm{MSA}(D_{i+1}))\cdot E_i + E_i. \tag{2}$$

Unlike previous works [14], [28], [52] that mainly use the addition operation or the concatenation operation to fuse the features from different feature levels, we first compute the

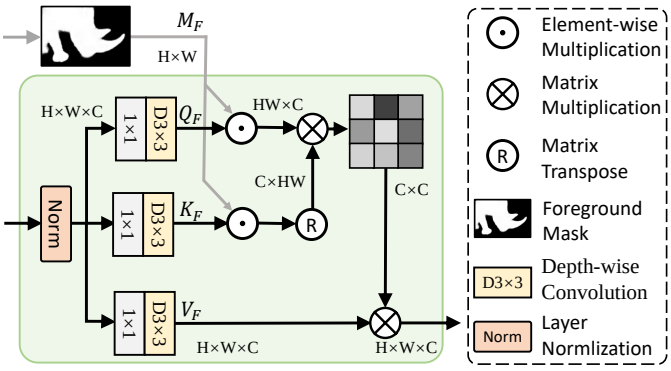


Fig. 3. Diagrammatic details of the proposed F-TA in our MSA. Our B-TA shares a similar structure except for the mask.

element-wise product between them and then use the summation operation. We empirically found that such a simple modification brings about 0.2%+ relative improvement in terms of S-measure and weighted F-measure averagely on NC4K [47], COD10K-test [13], and CAMO-test [34].

Loss Function. Following [24], [67], we add side supervision at each feature level. We denote the predictions generated by the decoder of CamoFormer as $\{P_i\}_{i=1}^{5}$. Except for the final prediction map P_1, all the other prediction maps P_i are used in the MSAs for the progressive refinement as described above. During training, each P_i is rescaled to the same size as the input image, and all of them are supervised by the BCE loss [9] and IoU loss [49]. Following [13], the overall loss is a summation of multi-stage loss. The total loss of our CamoFormer can be formulated as follows:

$$\mathcal{L}(P, G) = \sum_{i=1}^{5} \mathcal{L}_{bce}(P_i, G) + \mathcal{L}_{iou}(P_i, G), \tag{3}$$

where G is the ground truth annotation.

3.2 Masked Separable Attention

Camouflaged objects are diverse in scale and highly similar to the background, which makes them difficult to segment completely. How to accurately identify camouflaged objects from the background is crucial. We solve this by presenting the masked separable attention (MSA), where different attention heads take charge of different functionalities. We intend to use part of the attention heads to separately calculate the attention scores in the predicted foreground and background regions and use them to identify the camouflaged objects better.

Our MSA is based on a modified version of self-attention to save computations, namely Multi-Dconv Head Transposed Attention [71], which we denote as TA for short. Given an input $\mathbf{X} \in \mathbb{R}^{HW \times C}$ where H and W are respectively the height and width while C is the channel number, TA can be formulated as:

$$\text{TA}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{V} \cdot \text{Softmax}(\frac{\mathbf{Q}^\top \mathbf{K}}{\alpha}), \tag{4}$$

where $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are the query, key, and value matrices that can be generated by using three separate 1×1 convolutions followed by a 3×3 depthwise convolution, and α is a learnable scaling parameter. In practical use, Eqn. 4 can also be extended to a multi-head version, as done in the original self-attention [62], to augment the feature representations.

Masked Separable Attention. The attention heads in the above TA are equally utilized for encoding spatial information. Differently, in our MSA, we propose to introduce a prediction mask that can be generated at each feature level into TA as a foreground-background contrast prior for better recognizing the camouflaged objects. To achieve this, we divide all the attention heads into three groups: foreground-head TA (F-TA), background-head TA (B-TA), and the normal TA. The structural details of our MSA are shown in Fig. 3.

To be specific, given a predicted foreground mask M_F, the formulation of F-TA can be written as:

$$\text{F-TA}(\mathbf{Q}_F, \mathbf{K}_F, \mathbf{V}_F) = \mathbf{V}_F \cdot \text{Softmax}(\frac{\mathbf{Q}_F^\top \mathbf{K}_F}{\alpha_F}), \tag{5}$$

where $\mathbf{Q}_F$, $\mathbf{K}_F$ are the masked query and key matrices that can be produced by multiplying them with M_F and $\mathbf{V}_F$ is the value matrix without masking. In this way, the features can be refined by building pairwise relationships within the foreground regions, avoiding the influence of the background which may contain contaminative information. Similarly, given the background mask via the broadcast subtraction $M_B = 1 - M_F$, we can also conduct this process for the background. Thus, the formulation of B-TA can be written as:

$$\text{B-TA}(\mathbf{Q}_B, \mathbf{K}_B, \mathbf{V}_B) = \mathbf{V}_B \cdot \text{Softmax}(\frac{\mathbf{Q}_B^\top \mathbf{K}_B}{\alpha_B}). \tag{6}$$

Other than the F-TA heads and B-TA heads, the third group of the heads is kept unchanged as in Eqn. 4, which is used to build relationships between the foreground and background. The outputs of all the heads are then concatenated and sent into a 3×3 convolution for feature aggregation and map the number of channels to C_d:

$$\mathbf{Z} = \text{Conv}_{3\times3}([\text{F-TA}, \text{B-TA}, \text{TA}]), \tag{7}$$

where $[\cdots]$ is the concatenation operation.

Mask Generation. At each feature level, a mask should be generated by a 3×3 convolution followed by a Sigmoid function and then used in our MSA. As supervision is added to each feature level, we directly use the predictions $\{P_i\}_{i=2}^{5}$ as masks and sent each of them to the corresponding MSA. Note that we do not binarize the prediction maps but keep them as continuous maps ranging from 0 to 1, which we found works better in our experiments.

4 Measuring Border Regions

As mentioned in Sec. 1, camouflaged objects often share similar patterns with their surroundings, making the regions around the object borders difficult to recognize. In this section, we propose to use the weighted F-measure (wF) and mean absolute error (M) to evaluate the segmentation results around the camouflaged region boundaries, resulting in two new metrics: adaptive border region constrained weighted F-measure (BR-F) and mean absolute error (BR-M). The detailed calculation processes of these two metrics are described as follows.

Fig. 4. Illustration for the region of borders. 'αBR@15' and 'αBR@30' are the adaptive border regions that are generated by dilating the borders of GT via r with C=15 and r with C=30 dilation kernels respectively.

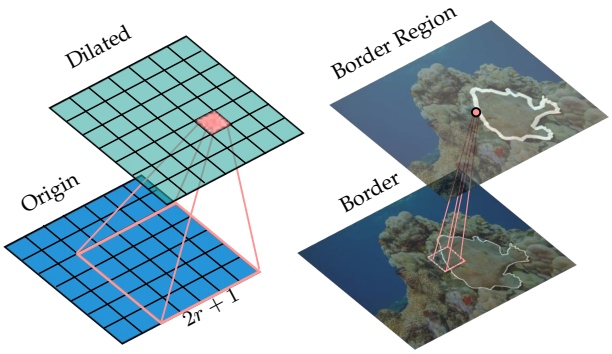


Fig. 5. Visualization of the dilation operation. Left: dilation operation at (i, j) with a dilation rate 5. Right: visualization of the dilation operation on an image.

Border Region Generation. Given the ground truth (GT), the borders $B \in \{0,1\}^{H\times W}$ are attained via the binary image border search method [59], as shown in Fig. 4. We attempt to measure on a relatively larger region, which is beneficial to observe the texture deviations of borders. The border regions are generated by dilating the GT borders, as shown in Fig. 5. To be specific, the value at location (i,j) after dilation can be written as:

$$BR(i,j) = \text{Max}(B((i,j),r)), \tag{8}$$

where $B((i,j),r)$ is a square region in the border map centered at location (i,j) with side length $2r+1$. Considering the variety of the target sizes, a border region with a fixed width is not suitable. Thus, we calculate the dilation rate r according to the object size:

$$r = \sum G/(\alpha \times H \times W), \tag{9}$$

where $G \in [0,1]^{H\times W}$ is the ground truth for the image and α is a scaling factor that scales r to around 1. We set it to 0.05. Then, $C \cdot r$ is the dilation width for the border region, where C is the base width and r is adjusted according to the samples.

The whole border region mask $BR \in \{0,1\}^{H\times W}$ can be generated by dilating the border via the corresponding dilation width $C \cdot r$ for each camouflaged region border location. We denote the adaptive border region with base dilation width C as 'αBR@C', and we visualize the border regions in Fig. 4.

Metric Calculation. For the metric BR-M, we calculate the M [54] value in the chosen border region. Given the border region mask BR, the BR-M can be defined as:

$$\text{BR-M} = \frac{\sum |P-G| \cdot BR}{\sum BR}, \tag{10}$$

where P and G are the prediction map and the corresponding GT, which have the same shape as BR.

Similar to BR-M, the metric BR-F is implemented by calculating the weighted F-measure [48] in the selected border regions. Following [48], the weighted error map E^w is calculated according to P and G. The four basic quantities, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in our BR-F can be defined as follows:

$$\begin{aligned} TP^{BR} &= BR \cdot (1-E^w) \cdot G, \\ TN^{BR} &= BR \cdot (1-E^w) \cdot (1-G), \\ FP^{BR} &= BR \cdot E^w \cdot (1-G), \\ FN^{BR} &= BR \cdot E^w \cdot G. \end{aligned} \tag{11}$$

Note that the difference between our quantities and the original ones is that our calculation process is confined to the border regions. The border region constrained weighted precision and recall can be written as:

$$\begin{aligned} \text{Precision}^{BR} &= \frac{TP^{BR}}{TP^{BR}+FP^{BR}}, \\ \text{Recall}^{BR} &= \frac{TP^{BR}}{TP^{BR}+FN^{BR}}. \end{aligned} \tag{12}$$

Finally, the BR-F measure is defined as:

$$\text{BR-F} = (1+\beta^2)\frac{\text{Precision}^{BR} \cdot \text{Recall}^{BR}}{\beta^2 \cdot \text{Precision}^{BR} + \text{Recall}^{BR}}. \tag{13}$$

5 Experimental Results

5.1 Experiment Setup

Implementation Details. We implement our CamoFormer using the Pytorch library [53]. A pretrained PVTv2 [65] on the ImageNet dataset [32] is employed as the encoder of our network. Unless otherwise specified, we adopt PVTv2 [65] as the backbone. Besides, we also report results with other backbones, e.g., Transformer-based Swin Transformer [42], CNN-based ResNet [23] and ConvNeXt [43]. SGD with momentum 0.9 and weight decay 2e-4 is used as the optimizer. The learning rate is initially set to 5e-3 and decays following the cosine learning rate strategy. During training, all the input images are resized to 384×384. The entire model is trained end-to-end for 60 epochs costing around 7 hours with a batch size of 6 on an NVIDIA V100 GPU.

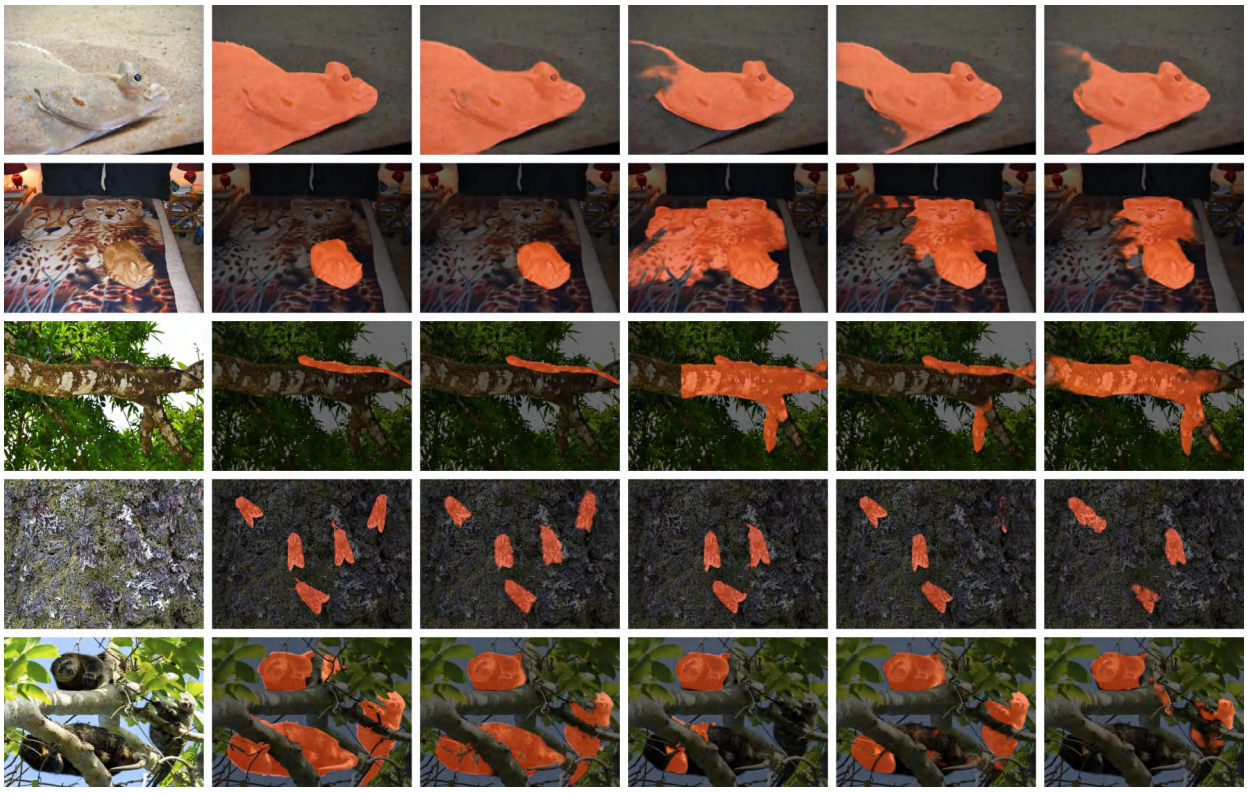


Fig. 6. Visualization comparisons between our CamoFormer and other SOTA methods. Segmentation results are shown in orange.

Datasets. We evaluate our methods on three popular COD benchmarks, including CAMO [34], COD10K [13], and NC4k [47]. CAMO comprises 2,500 images, half of which contain camouflaged objects and half do not. COD10k includes 5,066 camouflaged, 3,000 background, and 1,934 non-camouflaged images. NC4K is a large-scale COD dataset consisting of 4,121 images for testing. Following previous works [14], [28], [52], we use 1,000 images from the CAMO dataset and 3,040 images from COD10K for training and the others for testing.

Metrics. Following [28], [52], [79], we use four golden metrics for evaluation, including Structure-measure (S_m) [11], mean absolute error (M) [54], weighted F-measure (wF) [48], and adaptive E-measure (αE) [12]. M is the absolute difference between the prediction map and GT. S_m simultaneously evaluates region-aware and object-aware structural similarity between predictions and GT. wF is an exhaustive measure of both recall and precision. αE evaluates element-wise similarity and statistics at image level. Besides, we draw the precision-recall (PR) curves, F_β-threshold (F_β) curves in Fig. 8 and false negative ratios (FNRs) in Fig. 9.

5.2 Qualitative Evaluation

Visualization of Predictions. Fig. 6 presents the visualization samples of our CamoFormer and three previous SOTA methods. To better show the performance of these models, several typical samples containing different complex scenarios in the COD field are selected. As shown in the top row, other methods still struggle to precisely perceive the camouflaged objects from their similar surroundings. Sometimes, it is also difficult for them to identify the camouflaged objects owing to the lack of global contrast information, as shown in the bottom two rows. In short, they misjudge some regions or miss parts of the targets when dealing with complex conditions due to the lack of a comprehensive understanding of the foreground and background. In contrast, by explicitly perceiving the foreground-background clues, our CamoFormer can generate high-quality segmentation maps of the camouflaged objects even under difficult conditions.

Object Border Quality Comparison. Camouflaged objects sometimes possess peculiar-looking shapes, as shown in Fig. 7. To demonstrate how well our CamoFormer performs when coping with these kinds of objects, we show some prediction results in Fig. 7 and depict the GT object borders with white curves. The borders of our predictions are closer to those of the GT objects, while there are obvious deviations in the predictions by other methods. These visualizations indicate that our model can segment more precise camouflaged targets.

5.3 Quantitative Evaluation

We compare our CamoFormer with 12 CNN-based SOTA COD models, including ZoomNet [52], FDNet [79], SegMaR [28], DGNet [26], SINetV2 [13], C^2FNet [58], UJSC [35], PFNet [51], MGL-R [72], SLSR [47], SINet [14], and PraNet [16] and 6 Transformer-based methods, including COS-T [63], TPRNet [75], VST [41], DTINet [44], UGTR [68], and ICON [81]. For a fair comparison, the prediction results are directly provided by their authors or generated by their well-trained models.

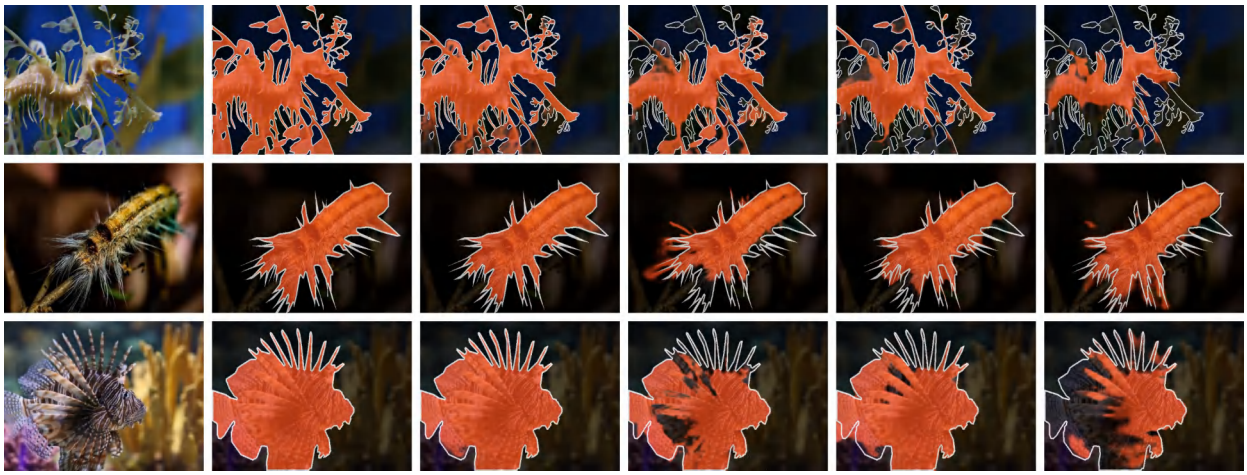


Fig. 7. Comparisons of our CamoFormer and other SOTA methods on the borders of segmentation. The borders of GT are marked in white, and the ones of predictions are in orange.

TABLE 1
Comparison of our CamoFormer with the recent SOTA methods. '-R': ResNet50 [23], '-C': ConvNext-base [43], '-S': Swin Transformer-base [42], '-P': PVTv2-b4 [65]. As can be seen, our CamoFormer-P performs much better than previous methods with either CNN- or Transformer-based models. '↑': the higher the better, '↓': the lower the better.

Method	NC4K (4,121)				COD10K-Test (2,026)				CAMO-Test (250)			
	S_m ↑	αE ↑	wF ↑	M↓	S_m ↑	αE ↑	wF ↑	M↓	S_m ↑	αE ↑	wF ↑	M↓
CNN-Based Methods												
PraNet$_{2020}$ [16]	0.822	0.871	0.724	0.059	0.789	0.839	0.629	0.045	0.769	0.833	0.663	0.094
SINet$_{2020}$ [14]	0.808	0.883	0.723	0.058	0.776	0.867	0.631	0.043	0.745	0.825	0.644	0.092
SLSR$_{2021}$ [47]	0.840	0.902	0.766	0.048	0.804	0.882	0.673	0.037	0.787	0.855	0.696	0.080
MGL-R$_{2021}$ [72]	0.833	0.893	0.739	0.053	0.814	0.865	0.666	0.035	0.782	0.847	0.695	0.085
PFNet$_{2021}$ [51]	0.829	0.892	0.745	0.053	0.800	0.868	0.660	0.040	0.782	0.852	0.695	0.085
UJSC$_{2021}$ [35]	0.842	0.907	0.771	0.047	0.809	0.891	0.684	0.035	0.800	0.853	0.728	0.073
C^{2}FNet$_{2021}$ [58]	0.838	0.898	0.762	0.049	0.813	0.886	0.686	0.036	0.796	0.864	0.719	0.080
SINetV2$_{2022}$ [13]	0.847	0.898	0.770	0.048	0.815	0.863	0.680	0.037	0.820	0.875	0.743	0.070
SegMaR$_{2022}$ [28]	0.841	0.905	0.781	0.046	0.833	0.895	0.724	0.033	0.815	0.872	0.742	0.071
ZoomNet$_{2022}$ [52]	0.853	0.907	0.784	0.043	0.838	0.893	0.729	0.029	0.820	0.883	0.752	0.066
FDNet$_{2022}$ [79]	0.834	0.895	0.750	0.052	0.837	0.897	0.731	0.030	0.844	0.903	0.778	0.062
DGNet$_{2023}$ [26]	0.857	0.907	0.784	0.042	0.822	0.877	0.693	0.033	0.839	0.901	0.769	0.057
CamoFormer-R (Ours)	0.857	0.915	0.793	0.041	0.838	0.898	0.730	0.029	0.817	0.884	0.756	0.066
CamoFormer-C (Ours)	0.884	0.936	0.833	0.033	0.860	0.923	0.767	0.024	0.860	0.920	0.811	0.051
Transformer-Based Methods												
COS-T$_{2021}$ [63]	0.825	0.881	0.730	0.055	0.790	0.901	0.693	0.035	0.813	0.896	0.776	0.060
VST$_{2021}$ [41]	0.830	0.887	0.740	0.053	0.810	0.866	0.680	0.035	0.805	0.863	0.780	0.069
UGTR$_{2021}$ [68]	0.839	0.886	0.746	0.052	0.817	0.850	0.666	0.036	0.784	0.859	0.794	0.086
ICON$_{2022}$ [81]	0.858	0.914	0.782	0.041	0.818	0.882	0.688	0.033	0.840	0.902	0.769	0.058
TPRNet$_{2022}$ [75]	0.854	0.903	0.790	0.047	0.829	0.892	0.725	0.034	0.814	0.870	0.781	0.076
DTINet$_{2022}$ [44]	0.863	0.915	0.792	0.041	0.824	0.893	0.695	0.034	0.857	0.912	0.796	0.050
CamoFormer-S (Ours)	0.888	**0.941**	0.840	0.031	0.862	0.932	0.772	0.024	0.876	**0.935**	0.832	**0.043**
CamoFormer-P (Ours)	**0.893**	0.940	**0.850**	**0.030**	**0.872**	**0.934**	**0.793**	**0.022**	**0.878**	0.934	**0.839**	0.044

Performance on Object Regions. As shown in Tab. 1, our proposed CamoFormer consistently and significantly surpasses the previous methods on all three benchmarks without any post-process tricks or extra data for training. Compared to the recent CNN-based COD methods, such as ZoomNet [52], FDNet [79], and SegMaR [28], although they adopt strategies, like multi-stage training and inference that cost extra computational burden, our CamoFormer still outperforms them on all benchmarks by a large margin. Meanwhile, compared to the Transformer-based models (e.g., TPRNet [75] and DTINet [44]), our method also performs better than them.

Performance on Border Regions. The borders of camouflaged targets are challenging to detect due to their irregular shapes and high similarity with their surroundings. To quantify the segmentation performance near the border regions, we calculate the BR-F and BR-M scores described in Sec. 4, respectively. We attain the border regions by dilating the boundaries of the GT objects, as shown in Fig. 4. Note that the area of the region depends on the kernel size of the dilation operation. Tab. 2 shows the performance calculated in the border regions 'αBR@15' and 'αBR@30'.

TABLE 2
Comparison of our CamoFormer with recent SOTA methods on 'αBR@15' and 'αBR@30'. '-R': ResNet [23], '-C': ConvNext [43], '-P': PVTv2 [65], '-S': Swin Transformer [42], '↑': the higher the better, '↓': the lower the better.

Method	NC4K (4,121)				COD10K-Test (2,026)				CAMO-Test (250)			
	αBR@15		αBR@30		αBR@15		αBR@30		αBR@15		αBR@30	
	$wF\uparrow$	M↓	$wF\uparrow$	M↓	$wF\uparrow$	BR-M↓	$wF\uparrow$	M↓	$wF\uparrow$	M↓	$wF\uparrow$	M↓
CNN-Based Methods												
PraNet$_{20}$ [16]	0.727	0.069	0.756	0.052	0.635	0.093	0.667	0.074	0.669	0.081	0.688	0.063
SINet$_{20}$ [14]	0.706	0.071	0.741	0.053	0.610	0.093	0.651	0.074	0.632	0.083	0.654	0.065
SLSR$_{21}$ [47]	0.746	0.062	0.777	0.046	0.653	0.084	0.689	0.066	0.678	0.076	0.703	0.059
MGL-R$_{21}$ [72]	0.721	0.067	0.755	0.050	0.637	0.085	0.676	0.067	0.660	0.079	0.685	0.061
PFNet$_{21}$ [51]	0.736	0.066	0.768	0.049	0.653	0.087	0.689	0.068	0.694	0.075	0.717	0.057
UJSC$_{21}$ [35]	0.755	0.061	0.785	0.045	0.670	0.081	0.705	0.063	0.712	0.070	0.737	0.053
C^{2}FNet$_{21}$ [58]	0.751	0.063	0.781	0.047	0.667	0.084	0.704	0.065	0.709	0.071	0.731	0.055
DGNet$_{23}$ [26]	0.762	0.061	0.795	0.045	0.669	0.086	0.708	0.067	0.745	0.065	0.776	0.047
SINetV2$_{22}$ [13]	0.754	0.063	0.788	0.046	0.664	0.087	0.703	0.068	0.734	0.069	0.759	0.051
SegMaR$_{22}$ [28]	0.748	0.060	0.783	0.044	0.723	0.070	0.755	0.054	0.737	0.065	0.762	0.049
ZoomNet$_{22}$ [52]	0.767	0.058	0.797	0.043	0.698	0.072	0.735	0.056	0.730	0.066	0.758	0.049
FDNet$_{22}$ [79]	0.726	0.071	0.765	0.051	0.668	0.088	0.715	0.066	0.745	0.067	0.777	0.048
CamoFormer-R (Ours)	0.768	0.057	0.800	0.042	0.696	0.071	0.740	0.055	0.725	0.066	0.755	0.048
CamoFormer-C (Ours)	0.806	0.050	0.838	0.035	0.725	0.070	0.765	0.052	0.789	0.055	0.817	0.039
Transformer-Based Methods												
VST$_{21}$ [41]	0.754	0.061	0.786	0.046	0.677	0.080	0.714	0.062	0.714	0.071	0.736	0.054
UGTR$_{21}$ [68]	0.725	0.068	0.763	0.050	0.629	0.088	0.673	0.069	0.670	0.079	0.698	0.061
ICON$_{22}$ [81]	0.742	0.068	0.785	0.048	0.639	0.097	0.682	0.075	0.732	0.071	0.769	0.051
TPRNet$_{22}$ [75]	0.758	0.061	0.789	0.045	0.677	0.081	0.712	0.064	0.718	0.070	0.741	0.053
DTINet$_{22}$ [44]	0.776	0.058	0.808	0.042	0.674	0.084	0.714	0.065	0.776	0.060	0.806	0.043
CamoFormer-S (Ours)	0.803	0.051	0.839	0.036	0.710	0.074	0.754	0.055	0.797	0.054	0.831	0.037
CamoFormer-P (Ours)	**0.819**	**0.047**	**0.850**	**0.033**	**0.741**	**0.066**	**0.779**	**0.050**	**0.805**	**0.051**	**0.835**	**0.036**

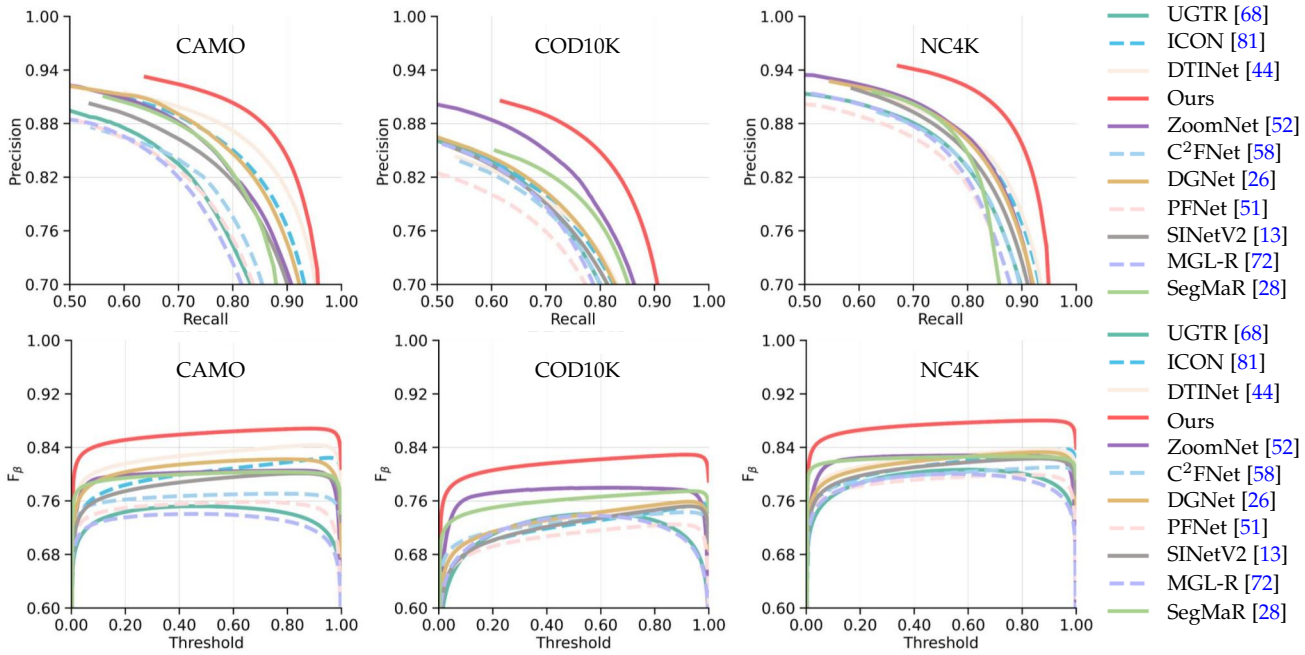


Fig. 8. PR and f_β curves of the proposed CamoFormer and the recent SOTA algorithms on all benchmarks.

Remarkably, CamoFormer achieves much better results than other methods, demonstrating that our predictions perform better at GT object boundaries, while the predictions of other methods are significantly biased.

PR & F_β curves of COD methods. We provide the PR and F_β curves of our CamoFormer and previous methods on NC4K, CAMO, and COD10K datasets, as shown in Fig. 8. Note that the higher the curve is, the better the model performs. It is clear that our CamoFormer (red curve) surpasses all other methods.

Computational Cost Comparison. Tab. 3 shows the parameters and MACs costed by our CamoFormer and the recent

TABLE 3
Comparisons of our CamoFormer and other SOTA methods on the number of parameters and MACs. The calculation is based on the same code and these methods are evaluated when keeping the inference settings in the corresponding papers. The inference time, *i.e.,* frames per second (FPS), is calculated on a single NVIDIA 3090 GPU.

Methods	Ours	DTINet [44]	ZoomNet [52]	UGTR [68]	C^2F-Net [58]	UJSC [35]	PFNet [51]	MGL-R [72]	SINet [14]
Params	71.3M	266.3M	33.4M	48.9M	28.4M	218.0M	46.5M	63.6M	48.9M
MACs	47.1G	145.6G	101.8G	500.1G	13.1G	56.2G	26.7G	277.0G	19.4G
FPS	41.3	19.7	24.0	16.6	65.8	34.2	62.6	13.4	56.5

TABLE 4
Ablation study of our CamoFormer variants. 'Baseline': the transformer backbone and several convolution layers; '+MSA': Baseline with MSA; 'w/TA only': Baseline equipped with TA and iterative refinement fashion.

Setting	NC4K (4,121)				COD10K-Test (2,026)				CAMO-Test (250)			
	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$
Baseline	0.859	0.916	0.801	0.043	0.830	0.904	0.719	0.032	0.838	0.891	0.780	0.058
Baseline+MSA	0.875	0.926	0.815	0.036	0.848	0.906	0.735	0.029	0.858	0.918	0.808	0.052
CamoFormer-P w/ TA only	0.871	0.920	0.808	0.039	0.844	0.908	0.741	0.029	0.859	0.910	0.810	0.055
CamoFormer-P	**0.893**	**0.940**	**0.850**	**0.030**	**0.872**	**0.934**	**0.793**	**0.022**	**0.878**	**0.934**	**0.839**	**0.044**

TABLE 5
Ablation study on the proposed MSA. All three branches ('TA', 'F-TA', and 'B-TA') contribute to the overall performance. In addition, eliminating either the 'F-TA' or 'B-TA' branch hurts the performance.

	Decoder			NC4K (4,121)				COD10K-Test (2,026)				CAMO-Test (250)			
Settings	TA	F-TA	B-TA	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$wF\uparrow$	M$\downarrow$
1				0.865	0.921	0.795	0.041	0.836	0.915	0.724	0.030	0.857	0.915	0.797	0.053
2	✓			0.871	0.920	0.808	0.039	0.844	0.908	0.741	0.029	0.859	0.910	0.810	0.055
3		✓		0.881	0.927	0.824	0.034	0.855	0.918	0.747	0.026	0.868	0.920	0.821	0.053
4			✓	0.879	0.924	0.819	0.035	0.851	0.915	0.742	0.028	0.856	0.914	0.809	0.055
5		✓	✓	0.887	0.937	0.844	0.032	0.869	0.930	0.783	0.024	0.875	0.933	0.838	0.045
6	✓		✓	0.885	0.933	0.839	0.033	0.860	0.924	0.763	0.025	0.866	0.926	0.827	0.048
7	✓	✓		0.889	**0.940**	0.845	0.031	0.866	0.927	0.773	0.024	0.871	0.929	0.832	0.046
8	✓	✓	✓	**0.893**	**0.940**	**0.850**	**0.030**	**0.872**	**0.934**	**0.793**	**0.022**	**0.878**	**0.934**	**0.839**	**0.044**

SOTA methods. As can be seen, our CamoFormer takes an acceptable computational cost compared to other methods but receives much better results.

False Negative Ratios (FNRs). We also adopt FNR [81] to intuitively show the superiority of our CamoFormer over other cutting-edge COD models. Note that the false negative metric refers to the pixels with wrong predictions on the target objects, and FNR [81] aims to calculate the proportion of mispredicted pixels on target objects to all pixels on those objects. A lower FNR means the model is better. To be specific, the formulation of FNR is defined as follows:

$$FN(x,y) = \begin{cases} 1, & G(x,y) = 1 \& P(x,y) = 0 \\ 0, & others \end{cases},$$
$$\text{FNR} = \frac{\sum_{x=1}^{W}\sum_{y=1}^{H} FN(x,y)}{\sum_{x=1}^{W}\sum_{y=1}^{H} G(x,y)}, \tag{14}$$

where FN is a pixel-level indicator to determine which pixels belong to false negatives, and G is the ground truth.

As shown in Fig. 9, our CamoFormer achieves the lowest FNR scores across all datasets, which demonstrates the efficiency of our model in capturing complete targets.

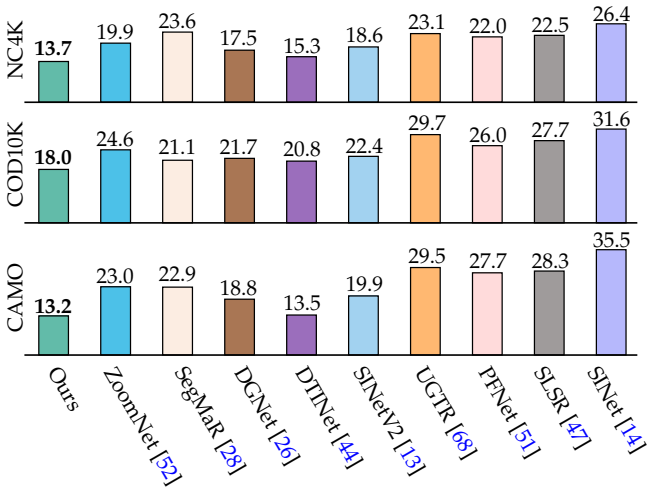


Fig. 9. FNR statistics of 10 methods on three different datasets. The best results are highlighted in red.

5.4 Method Analysis

Overall Results. We first ablate the network architecture of CamoFormer. The results are shown in Tab. 4. 'Baseline' refers to the model with only the encoder followed by a convolution for prediction. When our MSA is applied, the

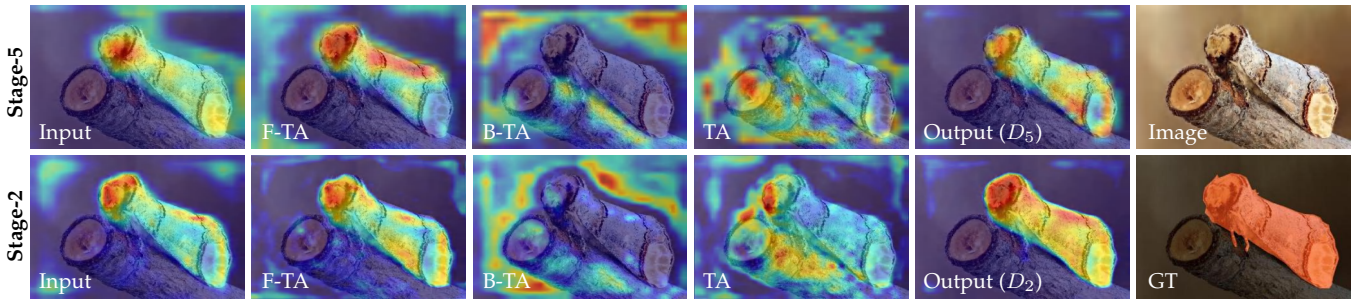


Fig. 10. Visualization of the feature maps around MSA. The features from Stage 2 and Stage 5 are chosen for comparison.

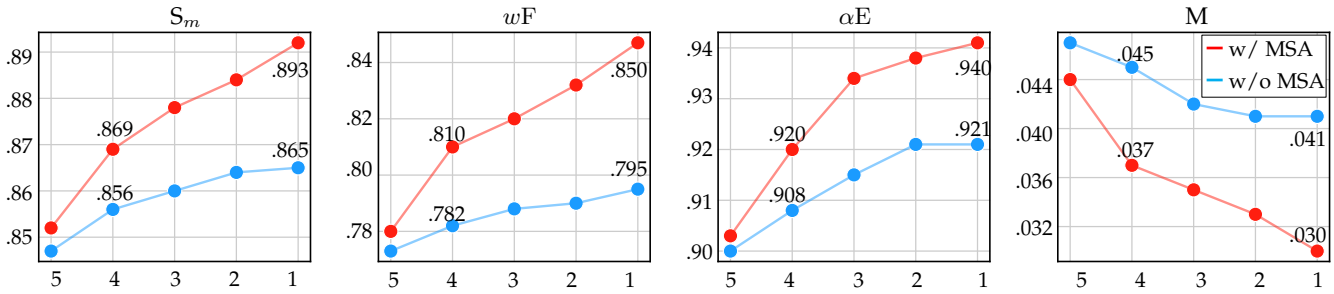


Fig. 11. Performance on the NC4K dataset at different feature levels of our progressive fusion strategy.

performance can be clearly improved in terms of all evaluation metrics compared to the 'Baseline'. Then, we attempt to add the decoder with only TA left. This progressive fusion strategy also helps compared to the 'Baseline'. Finally, we add our MSA in the decoder as in Fig. 2. We can see that the performance can be further improved. Both the top and bottom halves indicate the importance of MSA for COD.

Masked Separable Attention. We then ablate how each component in MSA helps. Tab. 5 shows the experimental results. The 'Baseline' in the first row can be viewed as a simple feature pyramid network, *i.e.*, no MSA is added in Fig. 2. We can observe that each type of attention component is helpful to improve performance. Though TA yields more performance gain compared to F-TA and B-TA, combining either F-TA or B-TA with TA can further improve the results. In particular, adding all three components yields the best results on all three datasets. This series of experiments indicates that separately processing the foreground and background with the proposed MSA is useful for segmenting camouflaged objects.

Feature Visualization. To provide more promising insights into our MSA, we also visualize the features around it. Note that the MSAs in Stages 2 and 5 are chosen for visualization. As shown in Fig. 10, F-TA and B-TA are able to effectively obtain the cues of the foreground and background. The features from F-TA, B-TA, and TA are complementary to form a complete camouflaged object. Consequently, the precise location and detailed information of the camouflaged object can be captured easily. In addition, compared to the features in Stage 5, we can distinguish the camouflaged targets more clearly according to the ones in Stage 2.

MSA in Progressive Fusion. In our model, a top-down path is built to progressively refine the segmentation map in the decoder. To show the impact of our MSA in this strategy, we depict the performance curves at different feature levels

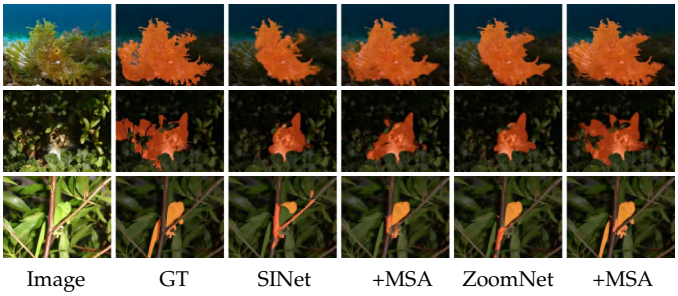


Fig. 12. Visualization results of SINet [14] and ZoomNet [52] equipped with the proposed MSA.

under the setup of w/ MSA and w/o MSA. The results can be found in Fig. 11. We can see that from feature level 5 to feature level 1, the performance gap between the model without MSA (blue line) and the one with MSA tends to be more significant for all four evaluation metrics. This indicates that the proposed MSA is compatible with the progressive fusion decoder.

Generalization of the MSA. To illustrate the generality of our MSA module, we apply it to some other COD methods, *i.e.*, SINet [14], and ZoomNet [52]. Specifically, for other methods, we generate the intermediary predictions using convolutions with 3×3 kernels and integrate our MSA module into each unit of their decoders. As shown in Tab. 6, our MSA brings consistent and significant improvements to other methods on all the camouflaged object detection benchmarks. We also present visualization results in Fig. 12. These results illustrate the generality of our method.

Decoder Width. The width (#channels) of the decoder affects not only the model size but also the inference speed. Tab. 7 shows the changes in model parameters, computational cost, and performance when the number of channels changes. We can see a clear improvement in our model

TABLE 6
Generalization ability of the MSA module. We apply our MSA module to SINet [14] and ZoomNet [52] and report the results.

Models	Computations		NC4K (4,121)				COD10K-Test (2,026)				CAMO (250)			
	Params	MAC	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$	$M \downarrow$	$S_m \uparrow$	$\alpha E \uparrow$	$wF \uparrow$	$M \downarrow$
SINet [14]	49M	19G	0.808	0.883	0.723	0.058	0.776	0.867	0.631	0.043	0.745	0.825	0.644	0.092
+MSA	57M	25G	0.829	0.900	0.752	0.052	0.809	0.883	0.675	0.035	0.790	0.844	0.681	0.077
ZoomNet [52]	33M	102G	0.853	0.907	0.784	0.043	0.838	0.893	0.729	0.029	0.820	0.883	0.752	0.066
+MSA	39M	111G	0.870	0.920	0.800	0.038	0.850	0.904	0.745	0.026	0.835	0.899	0.770	0.061

TABLE 7
Ablation study on the channel numbers in the decoder. All the variants are equipped with our MSA.

Settings	Computations		NC4K (4,121)				COD10K-Test (2,026)				CAMO (250)			
	Params	MAC	$S_m \uparrow$	αE	$wF \uparrow$	M	$S_m \uparrow$	αE	$wF \uparrow$	M	$S_m \uparrow$	αE	$wF \uparrow$	M
$C_d = 32$	63M	30G	0.890	0.939	0.844	0.031	0.867	0.929	0.782	0.024	0.873	0.928	0.831	0.046
$C_d = 64$	65M	34G	0.890	0.940	0.846	0.031	0.869	0.929	0.787	0.023	0.876	0.931	0.833	0.046
$C_d = 128$	71M	47G	**0.893**	0.940	0.850	**0.030**	0.872	**0.934**	**0.793**	**0.022**	0.878	**0.934**	**0.839**	**0.044**
$C_d = 192$	82M	69G	**0.893**	**0.942**	**0.851**	**0.030**	**0.873**	**0.934**	0.790	0.023	**0.879**	**0.934**	0.838	**0.044**
$C_d = 256$	97M	99G	0.892	**0.942**	0.846	**0.030**	0.870	0.932	0.790	0.023	0.876	0.931	0.837	0.045

when C_d increases from 32 to 128. However, when the width changes from 128 to 192, the performance improves little, but the parameters and computations rise. As a result, C_d is set to 128 for the trade-off between efficiency and model performance.

6 Conclusions

We present CamoFormer for camouflaged object segmentation. The core of our CamoFormer is the masked separable attention (MSA) that separately deals with the foreground and background regions using different attention heads. To make better use of our MSA, we adopt a progressive refinement decoder to gradually improve the segmentation quality at different feature levels in a top-down manner. Extensive experiments show that CamoFormer surpasses the existing 18 SOTA models with clear improvements.

References

[1] Yevgeny Beiderman, Mina Teicher, Javier Garcia, Vicente Mico, and Zeev Zalevsky. Optical technique for classification, recognition and identification of obscured objects. *Opt. Commun.*, 283(21):4274–4282, 2010.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Eur. Conf. Comput. Vis.*, 2020.

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[4] Gong Cheng, Jiabao Wang, Ke Li, Xingxing Xie, Chunbo Lang, Yanqing Yao, and Junwei Han. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.*, 60:1–11, 2022.

[5] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2014.

[6] Shupeng Cheng, Ge-Peng Ji, Pengda Qin, Deng-Ping Fan, Bowen Zhou, and Peng Xu. Large model based referring camouflaged object detection. *arXiv preprint arXiv:2311.17122*, 2023.

[7] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge. Implicit motion handling for video camouflaged object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[8] Hung-Kuo Chu, Wei-Hsin Hsu, Niloy J Mitra, Daniel Cohen-Or, Tien-Tsin Wong, and Tong-Yee Lee. Camouflage images. *ACM Trans. Graph.*, 29(4):51–1, 2010.

[9] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Ann. Oper. Res.*, 134(1):19–67, 2005.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2020.

[11] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Int. Conf. Comput. Vis.*, 2017.

[12] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Int. Joint Conf. on Artif. Intel.*, 2018.

[13] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10):6024–6042, 2022.

[14] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[15] Deng-Ping Fan, Ge-Peng Ji, Peng Xu, Ming-Ming Cheng, Christos Sakaridis, and Luc Van Gool. Advances in deep concealed scene understanding. *Visual Intelligence*, 1(1):16, 2023.

[16] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *Inter. Conf. on Med. Image Comp. and Computer-Assisted Interv.*, 2020.

[17] Meirav Galun, Eitan Sharon, Ronen Basri, and Achi Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Int. Conf. Comput. Vis.*, 2003.

[18] Shiming Ge, Xin Jin, Qiting Ye, Zhao Luo, and Qiang Li. Image editing by object-aware optimal boundary searching and mixed-domain composition. *Comp. Visual Media*, 4(1):71–82, 2018.

[19] Yuchao Gu, Lijuan Wang, Ziqin Wang, Yun Liu, Ming-Ming Cheng, and Shao-Ping Lu. Pyramid constrained self-attention network for fast video salient object detection. In *AAAI Conf. on Artif. Intel.*, 2020.

[20] Hongxing Guo, Yaling Dou, Ting Tian, Jingli Zhou, and Shengsheng Yu. A robust foreground segmentation method by temporal averaging multiple video frames. In *IEEE International conference on audio, language and image processing*, 2008.

[21] Joanna R Hall, Innes C Cuthill, Roland Baddeley, Adam J Shohet, and Nicholas E Scott-Samuel. Camouflage, detection and identification of moving targets. *Proc. Royal Soc. B*, 280(1758):20130064, 2013.

[22] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Process. Mag.*,

35(1):84–100, 2018.
[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[24] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):815–828, 2019.
[25] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
[26] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Deep gradient learning for efficient camouflaged object detection. *Mach. Intell. Research*, 20(1):92–108, 2023.
[27] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu. Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognition*, 123:108414, 2022.
[28] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
[29] Xinhao Jiang, Wei Cai, Zhili Zhang, Bo Jiang, Zhiyong Yang, and Xin Wang. Magnet: A camouflaged object detection network simulating the observation effect of a magnifier. *arXiv*, 2022. doi: https://doi.org/10.21203/rs.3.rs-1020529/v2.
[30] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. In *Adv. Neural Inform. Process. Syst.*, 2021.
[31] Ch Kavitha, B Prabhakara Rao, and A Govardhan. An efficient content based image retrieval using color and texture of image sub blocks. *Inter. Journal of Eng. Sci. and Tech.*, 3(2):1060–1068, 2011.
[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
[33] Chunbo Lang, Gong Cheng, Binfei Tu, Chao Li, and Junwei Han. Base and meta: A new perspective on few-shot segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023.
[34] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *Comp. Vis. and Image Understanding*, 184:45–56, 2019.
[35] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.
[37] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.
[38] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.*, 128(2):261–318, 2020.
[39] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[40] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.
[41] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Int. Conf. Comput. Vis.*, 2021.
[42] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, 2021.
[43] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
[44] Zhengyi Liu, Zhili Zhang, and Wei Wu. Boosting camouflaged object detection with dual-task interactive transformer. *Int. Conf. Pattern Recog.*, 2022.
[45] Xue-Jing Luo, Shuo Wang, Zongwei Wu, Christos Sakaridis, Yun Cheng, Deng-Ping Fan, and Luc Van Gool. Camdiff: camouflage image augmentation via diffusion. *CAAI Artificial Intelligence Research*, 2, 2023.
[46] Ziyang Luo, Nian Liu, Wangbo Zhao, Xuguang Yang, Dingwen Zhang, Deng-Ping Fan, Fahad Khan, and Junwei Han. Vscode: General visual salient and camouflaged object detection with 2d prompt learning. *arXiv preprint arXiv:2311.15011*, 2023.
[47] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[48] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.
[49] Gellért Máttyus, Wenjie Luo, and Raquel Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Int. Conf. Comput. Vis.*, 2017.
[50] Gerard Medioni. Generic object recognition by inference of 3-d volumetric. *Object Categorization: Computer and Human Vision Perspectives*, 87, 2009.
[51] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[52] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inform. Process. Syst.*, 2019.
[54] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.
[55] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.
[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Int. Conf. Learn. Represent.*, 2015.
[57] Martin Stevens and Sami Merilaita. Animal camouflage: current issues and new perspectives. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 364(1516):423–427, 2009.
[58] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. In *Int. Joint Conf. on Artif. Intel.*, 2021.
[59] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.
[60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.
[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Inter. Conf. Mach. Learning*, 2021.
[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.
[63] Haiwen Wang, Xinzhou Wang, Fuchun Sun, and Yixu Song. Camouflaged object segmentation with transformer. In *Cog. Sys. and Inform. Process.*, 2021.
[64] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Int. Conf. Comput. Vis.*, 2021.
[65] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Comp. Visual Media Journal*, 8(3):415–424, 2022.
[66] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Adv. Neural Inform. Process. Syst.*, 2021.
[67] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Int. Conf. Comput. Vis.*, 2015.
[68] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *Int. Conf. Comput. Vis.*, 2021.
[69] Bowen Yin, Xuying Zhang, Zhong-Yu Li, Li Liu, Ming-Ming Cheng, and Qibin Hou. Dformer: Rethinking rgbd representation learning for semantic segmentation. In *Proc. Int. Conf. Learn. Represent.*, 2024.
[70] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Trans.*

Pattern Anal. Mach. Intell., 2022. doi:https://doi.org/10.1109/TPAMI.2022.3206108.
[71] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
[72] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[73] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. In *Adv. Neural Inform. Process. Syst.*, 2021.
[74] Miao Zhang, Shuang Xu, Yongri Piao, Dongxiang Shi, Shusen Lin, and Huchuan Lu. Preynet: Preying on camouflaged objects. In *ACM Int. Conf. Multimedia*, 2022.
[75] Qiao Zhang, Yanliang Ge, Cong Zhang, and Hongbo Bi. Tprnet: camouflaged object detection via transformer-induced progressive refinement network. *The Visual Computer*, 39(10):4593–4607, 2023.
[76] Xuying Zhang, Bowen Yin, Zheng Lin, Qibin Hou, Deng-Ping Fan, and Ming-Ming Cheng. Referring camouflaged object detection. *arXiv preprint arXiv:2306.07532*, 2023.
[77] Xiang Zhang, Ce Zhu, Shuai Wang, Yipeng Liu, and Mao Ye. A bayesian approach to camouflaged moving object detection. *IEEE Trans. Circuit Syst. Video Technol.*, 27(9):2001–2013, 2016.
[78] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.
[79] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.
[80] Yupeng Zhou, Zhen Li, Chun-Le Guo, Song Bai, Ming-Ming Cheng, and Qibin Hou. Srformer: Permuted self-attention for single image super-resolution. In *Int. Conf. Comput. Vis.*, 2023.
[81] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. doi: https://doi.org/10.1109/TPAMI.2022.3179526.
[82] Mingchen Zhuge, Xiankai Lu, Yiyou Guo, Zhihua Cai, and Shuhan Chen. Cubenet: X-shape connection for camouflaged object detection. *Pattern Recognition*, 127:108644, 2022.

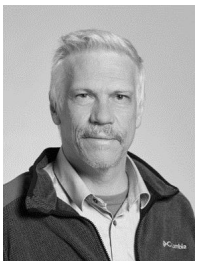
Bowen Yin is a Ph.D. student from the college of computer science, Nankai university. He is supervised by Prof. Qibin Hou. His research interests include computer vision and multimodal scene perception.

Xuying Zhang is a Ph.D. student from the college of computer science, Nankai university. He is supervised by Prof. Ming-Ming Cheng. His research interests include multimodal learning, camouflaged scene understanding, and 2D/3D visual perception.

Deng-Ping Fan (Senior Member, IEEE) will be joining the Department of Nankai International Advanced Research Institute (SHENZHEN-FUTIAN) as a faculty member in 2024. Now I'm a Full Professor and deputy director of the Media Computing Lab (MC Lab) at the College of Computer Science, Nankai University, China. Before that, he was postdoctoral, working with Prof. Luc Van Gool in Computer Vision Lab @ ETH Zurich. He is one of the core technique members in TRACE-Zurich project on automated driving.

Shaohui Jiao received her PhD degree from Chinese Academy of Sciences in 2010. She is now a researcher in MultiMedia Lab, Bytedance Inc. Her research interests include computer graphics, computer vision, VR, and AIGC.

Ming-Ming Cheng (Senior Member, IEEE) received his PhD degree from Tsinghua University in 2012, and then worked with Prof. Philip Torr in Oxford for 2 years. Since 2016, he is a full professor at Nankai University, leading the Media Computing Lab. His research interests include computer vision and computer graphics. He received awards, including ACM China Rising Star Award, IBM Global SUR Award, *etc*. He is a senior member of the IEEE and on the editorial boards of IEEE TPAMI and IEEE TIP.



Luc Van Gool is a professor at ETH Zurich and the head of the Computer Vision Laboratory (CV Lab). His main research interests include 2D and 3D object recognition, texture analysis, distance acquisition, stereo vision, robot vision, and optical flow. He has served as a member of the procedural committee for multiple top international conferences, including ICCV, ECCV, and CVPR. Received the David Marr Prize in 1998.

Qibin Hou (Member, IEEE) received his Ph.D. degree from the School of Computer Science, Nankai University. Then, he spent two wonderful years working at the National University of Singapore as a research fellow. Now, he is an associate professor at School of Computer Science, Nankai University. He has published more than 40 papers on top conferences/journals, including IEEE TPAMI, CVPR, ICCV, NeurIPS, etc. His research interests include deep learning and computer vision.