# Feature Shrinkage Pyramid for Camouflaged Object Detection with Transformers

Zhou Huang[1,2†] Hang Dai[3†] Tian-Zhu Xiang[4*] Shuo Wang[5]
Huai-Xin Chen[2] Jie Qin[6] Huan Xiong[7]
[1]Sichuan Changhong Electric Co., Ltd. [2]UESTC [3]University of Glasgow
[4]G42 [5]ETH Zurich [6]CCST, NUAA [7]MBZUAI
chowhuang@std.uestc.edu.cn, hang.dai@glasgow.ac.uk, {tianzhu.xiang19,
shawnwang.tech, qinjiebuaa, huan.xiong.math}@gmail.com, huaixinchen@uestc.edu.cn

## Abstract

*Vision transformers have recently shown strong global context modeling capabilities in camouflaged object detection. However, they suffer from two major limitations: less effective locality modeling and insufficient feature aggregation in decoders, which are not conducive to camouflaged object detection that explores subtle cues from indistinguishable backgrounds. To address these issues, in this paper, we propose a novel transformer-based Feature Shrinkage Pyramid Network (FSPNet), which aims to hierarchically decode locality-enhanced neighboring transformer features through progressive shrinking for camouflaged object detection. Specifically, we propose a non-local token enhancement module (NL-TEM) that employs the non-local mechanism to interact neighboring tokens and explore graph-based high-order relations within tokens to enhance local representations of transformers. Moreover, we design a feature shrinkage decoder (FSD) with adjacent interaction modules (AIM), which progressively aggregates adjacent transformer features through a layer-by-layer shrinkage pyramid to accumulate imperceptible but effective cues as much as possible for object information decoding. Extensive quantitative and qualitative experiments demonstrate that the proposed model significantly outperforms the existing 24 competitors on three challenging COD benchmark datasets under six widely-used evaluation metrics. Our code is publicly available at* https://github.com/ZhouHuang23/FSPNet.

## 1. Introduction

Camouflage is a common defense or tactic in organisms that "perfectly" blend in with their surroundings to deceive predators (prey) or sneak up on prey (hunters). Camouflaged object detection (COD) [11] aims to segment camouflaged objects in the scene and has been widely applied in species conservation [29], medical image segmentation [5, 20], and industrial defect detection [3], etc.

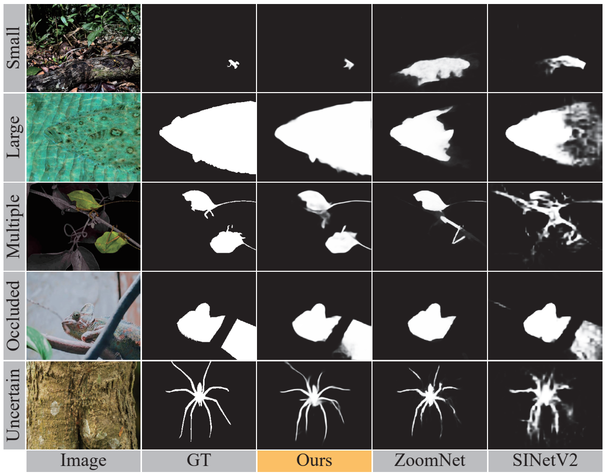

Figure 1. **Visual comparison of COD in different challenging scenarios**, including small, large, multiple, occluded and boundary-uncertain camouflaged objects. Compared with the recently proposed *ZoomNet* [30] and *SINet-v2* [10], our method provides superior performance with more accurate object localization and more complete object segmentation, mainly due to the proposed locality-enhanced global context exploration and progressive shrinkage decoder.

Due to the high similarities between camouflaged objects and their backgrounds, camouflaged objects are usually inconspicuous and indistinguishable, which brings great challenges to accurate detection. Recently, the development of deep learning and the availability of large-scale COD datasets (*e.g.*, COD10K [11]) have significantly advanced camouflaged object detection. Numerous deep learning-based methods have been proposed, which can be roughly divided into three categories: targeted design of feature exploration modules, multi-task joint learning frameworks,

[†]Equal contributions. *Corresponding author: *Tian-Zhu Xiang*.

and bio-inspired methods. Although these methods have made remarkable progress, they mainly rely heavily on convolutional neural networks (CNNs), which cannot capture long-range dependencies due to the limited receptive fields, resulting in inferior performance for COD. As shown in Fig. 1, recently proposed state-of-the-art CNN-based methods (*e.g.*, ZoomNet [30] and SINet-v2 [10]) fail to explore global feature relations and thus often provide predictions of incomplete object regions, especially for multiple objects, large objects and occlusion cases. Although larger convolution kernels or simply stacking multiple convolution layers with small kernels can enlarge receptive fields and thus alleviate this issue to some extent, it also dramatically increases the computational cost and the number of network parameters. Furthermore, studies [34] have shown that simply network deepening is ineffective for long-range dependency modeling.

Compared to CNNs, vision transformers (ViT) [7], which have recently been introduced into computer vision and demonstrated significant breakthroughs in various vision applications [17], can efficiently model long-range dependencies with the self-attention operations and thus overcome the above drawbacks of CNNs-based models. Recently, the works of [47] and [24] have attempted to accommodate transformers for COD and shown promising performance. These methods either employ transformer as a network component for feature decoding or utilize the off-the-shelf vision transformers as backbones for feature encoding. Through a thorough analysis of these methods for COD, we observe two major issues within existing techniques: 1) *Less effective local feature modeling for transformer backbones.* We argue that both global context and local features play essential roles in COD tasks. However, we observe that most transformer-based methods lack a locality mechanism for information exchange within local regions. 2) *Limitations of feature aggregation in decoders.* Existing decoders (shown in Fig. 2 (a)-(d)) usually directly aggregate the features with significant information differences (*e.g.*, low-level features with rich details and high-level features with semantics), which tends to discard some inconspicuous but valuable cues or introduce noise, resulting in inaccurate predictions. This is a big blow for the task of identifying camouflaged objects from faint clues.

To this end, in this paper, we propose a novel transformer-based Feature Shrinkage Pyramid Network, named *FSPNet*, which aims to hierarchically decode neighboring transformer features which are locality-enhanced global representations for camouflaged objects through progressive shrinking, thereby excavating and accumulating rich local cues and global context of camouflaged objects in our encoder and decoder for accurate and complete camouflaged object segmentation. Specifically, to complement local feature modeling in the transformer encoder, we propose a non-local token enhancement module (NL-TEM) which employs the non-local mechanism to interact neighboring similar tokens and explore graph-based high-level relations within tokens to enhance local representations. Furthermore, we design a feature shrinkage decoder (FSD) with adjacent interaction modules (AIMs) which progressively aggregates adjacent transformer features in pairs through a layer-by-layer shrinkage pyramid architecture to accumulate subtle but effective details and semantics as much as possible for object information decoding. Owing to the global context modeling of transformers, locality exploration within tokens and progressive feature shrinkage decoder, our proposed model achieves state-of-the-art performance and provides an accurate and complete camouflaged object segmentation. Our main contributions are summarized as follows:

- We propose a non-local token enhancement module (NL-TEM) for feature interaction and exploration between and within tokens to compensate for locality modeling of transformers.
- We design a feature shrinkage decoder (FSD) with the adjacent interaction module (AIM) to better aggregate camouflaged object cues between neighboring transformer features through progressive shrinking for camouflaged object prediction.
- Comprehensive experiments show that our proposed FSPNet achieves superior performance on three widely-used COD benchmark datasets compared to 24 existing state-of-the-art methods.

## 2. Related Work

### 2.1. CNN-based Camouflaged Object Detection

Recently, CNN-based approaches have made impressive progress on the COD task by releasing large-scale datasets. Some works attempt to mine inconspicuous features of camouflage objects from the background through meticulously designed feature exploration modules, *e.g.*, contextual feature learning [28, 36], texture-aware learning [60], and frequency-domain learning [57]. There are also some models [19, 21, 47] which propose to model uncertainty in data labeling or camouflaged data itself for COD. Besides, the multi-task learning framework is commonly used for COD. These methods generally introduce auxiliary tasks such as classification [18], edge/boundary detection [37,48,59], and object ranking [25]. Furthermore, some methods detect camouflaged objects by mimicking behavior patterns or visual mechanics of predators such as the search and identification process [10], and zooming in and out [16, 30]. In addition to image-based COD, more recently, [6] proposed to discover camouflaged objects in videos using motion information. Although CNN-based

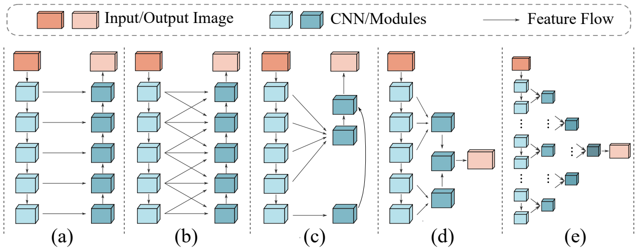

Figure 2. **Different types of decoding structures for object segmentation.** (a) U-shaped decoding structure [30, 35, 58]. (b) Dense integration strategy [31, 52, 53]. (c) Feedback refinement strategy [48, 55, 59]. (d) Separate decoding of low-level and high-level features [11, 12, 15]. (e) Our decoding structure.

models have achieved promising performance, these methods do not explore long-range dependencies due to limited receptive fields, which is critical for COD in images containing diverse objects.

### 2.2. Decoding Strategy

By reviewing vision tasks related to COD (*e.g.*, salient object detection and medical image segmentation), the decoder design can be summarized into four typical feature decoding strategies: (a) U-shaped decoding structure, (b) dense integration strategy, (c) feedback refinement strategy, and (d) separate decoding of low-level and high-level features, as shown in Fig. 2. Specifically, as the most prevalent feature decoding strategy, the U-shaped decoders [30, 35, 58] integrate lateral output multi-scale backbone features and recover object details gradually in a bottom-up manner. To weaken the interference of large resolution differences on the compatibility of feature fusion, some methods [31, 52, 53] use a dense integration strategy to aggregate multi-level features. Some methods treat the high-level output features separately (usually the last layer of backbone features) and then integrate them with other lateral outputs to improve localization and segmentation results. Some other methods [11, 12, 15] deal with low-level and high-level features differently to explore and integrate local cues and global semantics for object segmentation. Unlike the mainstream decoding strategy mentioned above, we adopt a pyramidal shrinkage decoding strategy, as shown in Fig. 2 (e), which aggregates adjacent features and recovers the object information layer by layer in a progressive manner.

### 2.3. Vision Transformer

Transformers, which are initially designed for natural language processing [41], have been widely applied in computer vision in recent years and achieved significant progress in numerous visual applications, such as image classification [7], object detection [4], and semantic segmentation [56]. Benefiting from the self-attention mechanism, transformers are better at capturing long-range dependencies when compared to CNN-based models [38, 43]. To our knowledge, ViT [7] is the first transformer model in computer vision community, which directly takes sequences of image patches as input to explore long-range spatial correlations for the classification task. Then a series of improved versions sprung up, such as data-efficient image transformers (DeiT) [40], pyramid vision transformer [42], and Swin transformer [23]. For camouflaged object detection, [32] propose a one-stage transformer framework for camouflaged instance segmentation. [47], [24], and [14] have made some attempts to detect camouflaged objects using transformers and achieved good performance. However, these methods remain limitations in the exploration of locality modeling and feature aggregation of decoders inherited from the CNN design paradigm, *i.e.*, information loss caused by large-span aggregation. In this paper, we design a feature shrinkage decoder with the adjacent interaction module to progressively aggregate adjacent features through the shrinkage pyramid for accurate decoding.

## 3. Proposed Method

### 3.1. Overview

Fig. 3 illustrates the overall architecture of our proposed FSPNet model. The main components include a vision transformer encoder, a non-local token enhancement module (NL-TEM), and a feature shrinkage decoder (FSD). Specifically, the input image is first serialized into tokens as input to a transformer encoder to model global contexts using the self-attentive mechanism. After that, to strengthen the local feature representation within tokens, a non-local token enhancement module (NL-TEM) is designed to perform feature interaction and exploration between and within tokens and convert the enhanced tokens from the encoder space to the decoder space for decoding. In the decoder, to merge and retain subtle but critical cues as much as possible, we design a feature shrinkage decoder (FSD) to progressively aggregates adjacent features through layer-by-layer shrinkage to decode object information.

### 3.2. Transformer Encoder

Unlike previous works on COD, we utilize a vanilla vision transformer (ViT) as the encoder to model the global context of camouflaged objects, mainly consisting of image serialization and transformer layers. a) Serialization. In order to satisfy the self-attention requirement on the input and reduce the computational complexity, inspired by [7], the given image $\boldsymbol{I} \in \mathbb{R}^{C\times H\times W}$ is first split into a sequence of non-overlapping image patches with patch size $(s, s)$, where $C$, $H$ and $W$ denote channel size, height and width of image $\boldsymbol{I}$, respectively, and $s = 16$ in our experiments. Then the image patches are linearly projected into a 1D sequence of token embeddings $T^0 \in \mathbb{R}^{l\times d}$, where $l = HW/s^2$ is the sequence length and $d = s^2 \cdot C$ is the embedding dimension.

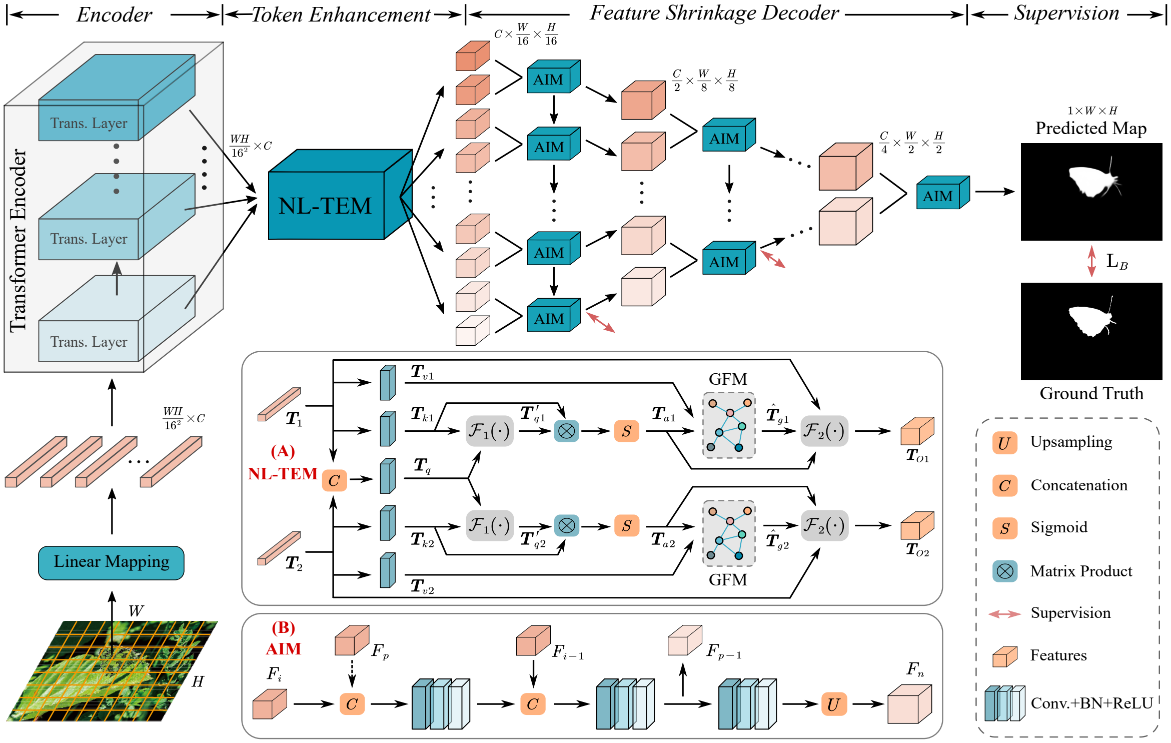

Figure 3. **Overall architecture of the proposed FSPNet.** It consists of three key components: a ViT-based encoder, a non-local token enhancement module (NL-TEM) and a feature shrinkage decoder (FSD) with adjacent interaction modules (AIM).

b) Transformer layer. To preserve positional information, an additional learnable position embedding $E^p$ is added to the tokens, forming the new tokens $T^p = T^0 + E^p$. All the tokens are then input into a transformer encoder with $n$ transformer layers, where each layer contains a multi-head self-attention (MSA) and a multi-layer perceptron (MLP) block. It can be formulated as:

$$\boldsymbol{T} = \mathrm{MLP}(\mathrm{MSA}(\boldsymbol{T}^p)), \tag{1}$$

where $\boldsymbol{T} \in \mathbb{R}^{l\times c}$, $c$ is the token dimension. Note that layer normalization [2] is applied before each block and residual connections after each block. Thus, we obtain the output tokens from the encoder.

### 3.3. Non-local Token Enhancement Module

Transformers bring powerful global context modeling capabilities but lack a locality mechanism for information exchange within a local region. Besides, it is well known that camouflaged targets always share very similar appearance information with noise objects and background, where the slight differences are difficult to be distinguished by low-order relations. Inspired by [39, 43], we design a non-local token enhancement module (NL-TEM) which is applied on neighboring tokens (local region) to strengthen the local feature representation. A non-local operation is first adopted to interact adjacent similar tokens for aggregation of adjacent camouflage clues. Then a graph convolution network (GCN) operation is employed to explore higher-order semantic relations between different pixels within tokens to spot subtle discriminative features. Specifically, as shown in Fig. 3 (A), given two adjacent tokens $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ from the transformer encoder, they are first normalized. Taking $\boldsymbol{T}_1$ as an example, it is passed through two linear projection functions (*i.e.*, $\omega_v$ and $\omega_k$), respectively, to obtain the dimension-reduced feature sequences $\boldsymbol{T}_v$ and $\boldsymbol{T}_k$ ($\in \mathbb{R}^{l\times\frac{c}{2}}$), which can be denoted as $\boldsymbol{T}_v = \omega_v(\boldsymbol{T}_1)$ and $\boldsymbol{T}_k = \omega_k(\boldsymbol{T}_1)$.

Besides, $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are concatenated to obtain an integrated token $\boldsymbol{T}_q$, which aggregates the features of both tokens, and is then exploited to interact with respective input tokens for feature enhancement. Specifically, another linear projection function $w_q$ is performed on this token with a dimension reduction of $c/2$, and then a softmax function is adopted to produce a weight map $T^w_q$. Next, the map is employed to weight $\boldsymbol{T}_k$ by element-wise multiplication, followed by an adaptive averaging pooling operation ($\mathcal{P}(\cdot)$) to reduce the computational costs. The above set of operations $\mathcal{F}_1(\cdot)$ can be denoted as:

$$T'_q = \mathcal{F}_1(\boldsymbol{T}_k, \boldsymbol{T}_q) = \mathcal{P}(\boldsymbol{T}_k \odot \mathrm{softmax}(w_q(\boldsymbol{T}_q))), \tag{2}$$

Then, the matrix product is applied to $\boldsymbol{T}_k$ and $\boldsymbol{T}'_q$ to explore correlations between the two, and a softmax operation is used to generate an attention map $\boldsymbol{T}_a$, which is denoted as

$\boldsymbol{T}_a = \mathrm{softmax}(\boldsymbol{T}'_q \otimes \boldsymbol{T}_k^\top)$.

After that, similar to [39], we feed the interactive token $\boldsymbol{T}_a$ and the token $\boldsymbol{T}_v$ into the graph fusion module (GFM). In GFM, $\boldsymbol{T}_v$ is projected into the graph domain by the attention mapping $\boldsymbol{T}_a$, denoted as $\boldsymbol{T}_g = \boldsymbol{T}_v \otimes \boldsymbol{T}_a^\top$. In this process, a collection of pixels ("regions") with similar features are projected to one vertex, and a single-layer GCN is adopted to learn high-level semantic relations between regions and reason over non-local regions to capture global representations within tokens, by cross-vertex information propagation on the graph. Specifically, the vertex features $\boldsymbol{T}_g$ are fed into the first-order approximation of the spectral graph convolution, and we can obtain the output $\hat{\boldsymbol{T}}_g$:

$$\hat{\boldsymbol{T}}_g = \mathrm{ReLU}((\boldsymbol{I} - \boldsymbol{A})\boldsymbol{T}_g w_g), \tag{3}$$

where $\boldsymbol{A}$ is the adjacency matrix of the encoded graph connectivity and $w_g \in \mathbb{R}^{16\times16}$ is the weight of the GCN.

Finally, a skip connection is used to combine the input token $\boldsymbol{T}_1$ with the graph-based enhanced representation, and then a deserialization ($\mathcal{D}(\cdot)$) operation is utilized to convert the token sequences to 2D image features with the same dimension as the original features for decoding, shown as:

$$\boldsymbol{T}_{O1} = \mathcal{F}_2(\hat{\boldsymbol{T}}_g, \boldsymbol{T}_a, \boldsymbol{T}_1) = \mathcal{D}(\hat{\boldsymbol{T}}_g \otimes \boldsymbol{T}_a^\top + \boldsymbol{T}_1), \tag{4}$$

where $\boldsymbol{T}_{O1} \in \mathcal{R}^{C\times\frac{H}{s}\times\frac{W}{s}}$ is the output local enhancement features from tokens. Similarly, we can also get $\boldsymbol{T}_{O2}$.

### 3.4. Feature Shrinkage Decoder

Common decoders, as shown in Fig. 2 (a)-(d), usually directly aggregate features with significant inconsistencies, *e.g.*, low-level features with rich details and high-level features with semantics, which easily introduces noise and loses subtle but valuable cues [26]. This is very unfriendly for the task of identifying camouflaged objects from inconspicuous cues. To this end, we design a feature shrinkage decoder (FSD) that progressively aggregates adjacent features in pairs using a hierarchical shrinkage pyramid architecture to accumulate more imperceptible effective cues. Furthermore, in our FSD decoder, we propose an adjacent interaction module (AIM) that interacts and merges the current adjacent feature pair and the aggregated features output by the previous AIM, and passes the current aggregated features to the next layer and the next AIM. It can be seen that AIM is served as a bridge for adjacent feature fusion and information passing (at the same layer and cross layer) in the decoder. As shown in Fig. 3, we can see that our decoder builds both bottom-up and left-to-right feature flows to retain more useful features. The proposed decoder can smoothly flow and accumulate the camouflaged object cues and avoid interference caused by large feature differences.

Specifically, suppose that $\boldsymbol{F}_i$ and $\boldsymbol{F}_{i-1}$ are the adjacent feature pair of the current layer, and $\boldsymbol{F}_p$ is the output aggregated feature from the previous AIM, AIM can be formulated as:

$$\begin{aligned} \boldsymbol{F}_p &= \mathrm{CBR}(\mathrm{Cat}(\mathrm{CBR}(\mathrm{Cat}(\boldsymbol{F}_{p-1}, \boldsymbol{F}_i)), \boldsymbol{F}_{i-1})) \\ \boldsymbol{F}'_i &= \mathrm{Up}(\mathrm{CBR}(\boldsymbol{F}_p)), \end{aligned} \tag{5}$$

where $\boldsymbol{F}_p$ is the feature passed to next AIM, and $\boldsymbol{F}'_i$ is the output feature of current AIM for next layer. $\mathrm{CBR}(\cdot)$ is composed of convolution, batch normalization, and ReLU operations. $\mathrm{Cat}(\cdot)$ and $\mathrm{Up}(\cdot)$ are the concatenation and 2× upsampling operations, respectively.

Note that FSD contains a total of 4 layers of shrinkage pyramid and 12 AIMs. The whole FSD process is summarized in Algorithm 1 in the *Supplementary Material*. The output feature from the last AIM is supervised by the ground truth ($\boldsymbol{G}$) after sigmoid and upsampling operations for camouflaged object prediction. We also supervise the output prediction ($\boldsymbol{P}_i$) at each layer of the FSD using a binary cross-entropy loss ($\mathcal{L}_{bce}$) and assign smaller weights to shallow outputs with lower detection precision. Finally, the overall loss function is:

$$\mathcal{L}_{total} = \sum_{i=0}^{2} 2^{(i-4)}\, \mathcal{L}_{bce}(\boldsymbol{P}_i, \boldsymbol{G}) + \mathcal{L}_{bce}(\boldsymbol{P}_3, \boldsymbol{G}), \tag{6}$$

where $i$ denotes the $i$-th layer of FSD and $P_3$ means the last layer of output prediction.

It should be noted that, unlike [26], the proposed FSD not only adopts the cross-layer feature interaction, but also adopts the feature interaction within the same layer, to better flow and accumulate effective features in the pyramid structure, thereby minimizing the loss of subtle but crucial features in the decoder process. Furthermore, we apply lateral supervision to each layer to force each decoder layer to mine and aggregate effective camouflaged object features. Besides, to alleviate the decoder structure, the proposed decoder only integrates adjacent features without overlapping, thus reducing aggregation operations. Tab. 3 shows the performance superiority of the proposed decoder.

## 4. Experiments and Results

### 4.1. Experiment Settings

**Datasets.** We evaluate the proposed method on three widely used COD datasets *i.e.*, CAMO [18], COD10K [11], and NC4K [25]. CAMO is the first COD dataset, containing 1,250 camouflaged images and 1,250 non-camouflaged images. COD10K is currently the largest COD dataset, which contains 5,066 camouflaged, 3,000 background, and 1,934 non-camouflaged images. NC4K is another recently released large-scale COD testing dataset which contains 4,121 images.

**Evaluation Metrics.** We adopt six well-known evaluation metrics, including S-measure [8] ($S_m$), weighted F-measure

Table 1. Quantitative comparison with 24 SOTA methods on three benchmark datasets. Notes ↑ / ↓ denote the larger/smaller is better, respectively. "–" is not available. The best and second best are **bolded** and underlined for highlighting, respectively.

| Methods | CAMO (250) | | | | | | COD10K (2,026) | | | | | | NC4K (4,121) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_m \uparrow$ | $F_\beta^\omega \uparrow$ | $F_\beta^m \uparrow$ | $E_\phi^m \uparrow$ | $E_\phi^x \uparrow$ | $\mathcal{M} \downarrow$ | $S_m \uparrow$ | $F_\beta^\omega \uparrow$ | $F_\beta^m \uparrow$ | $E_\phi^m \uparrow$ | $E_\phi^x \uparrow$ | $\mathcal{M} \downarrow$ | $S_m \uparrow$ | $F_\beta^\omega \uparrow$ | $F_\beta^m \uparrow$ | $E_\phi^m \uparrow$ | $E_\phi^x \uparrow$ | $\mathcal{M} \downarrow$ |
| Salient Object Detection | | | | | | | | | | | | | | | | | | |
| BASNet$_{19}$ | .618 | .413 | .475 | .661 | .708 | .159 | .634 | .365 | .417 | .678 | .735 | .105 | .695 | .546 | .610 | .762 | .786 | .095 |
| CPD$_{19}$ | .716 | .556 | .618 | .723 | .796 | .113 | .750 | .531 | .595 | .776 | .853 | .053 | .717 | .551 | .597 | .724 | .793 | .092 |
| EGNet$_{19}$ | .662 | .495 | .567 | .683 | .780 | .125 | .733 | .519 | .583 | .761 | .836 | .055 | .767 | .626 | .689 | .793 | .850 | .077 |
| SCRN$_{19}$ | .779 | .643 | .705 | .797 | .850 | .090 | .789 | .575 | .651 | .817 | .880 | .047 | .830 | .698 | .757 | .854 | .897 | .059 |
| F$^3$Net$_{20}$ | .711 | .564 | .616 | .741 | .780 | .109 | .739 | .544 | .593 | .795 | .819 | .051 | .780 | .656 | .705 | .824 | .848 | .070 |
| CSNet$_{20}$ | .771 | .642 | .705 | .795 | .849 | .092 | .778 | .569 | .635 | .810 | .871 | .047 | .750 | .603 | .655 | .773 | .793 | .088 |
| SSAL$_{20}$ | .644 | .493 | .579 | .721 | .780 | .126 | .668 | .454 | .527 | .768 | .789 | .066 | .699 | .561 | .644 | .780 | .812 | .093 |
| ITSD$_{20}$ | .750 | .610 | .663 | .780 | .830 | .102 | .767 | .557 | .615 | .808 | .861 | .051 | .811 | .680 | .729 | .845 | .883 | .064 |
| UCNet$_{20}$ | .739 | .640 | .700 | .787 | .820 | .094 | .776 | .633 | .681 | .857 | .867 | .042 | .811 | .729 | .775 | .871 | .886 | .055 |
| VST$_{21}$ | .787 | .691 | .738 | .838 | .866 | .076 | .781 | .604 | .653 | .837 | .877 | .042 | .831 | .732 | .771 | .877 | .901 | .050 |
| Camouflaged Object Detection | | | | | | | | | | | | | | | | | | |
| SINet$_{20}$ | .751 | .606 | .675 | .771 | .831 | .100 | .771 | .551 | .634 | .806 | .868 | .051 | .808 | .723 | .769 | .871 | .883 | .058 |
| SLSR$_{21}$ | .787 | .696 | .744 | .838 | .854 | .080 | .804 | .673 | .715 | .880 | .892 | .037 | .840 | .766 | .804 | .895 | .907 | .048 |
| PFNet$_{21}$ | .782 | .695 | .746 | .842 | .855 | .085 | .800 | .660 | .701 | .877 | .890 | .040 | .829 | .745 | .784 | .888 | .898 | .053 |
| MGL-R$_{21}$ | .775 | .673 | .726 | .812 | .842 | .088 | .814 | .666 | .711 | .852 | .890 | .035 | .833 | .740 | .782 | .867 | .893 | .052 |
| UJSC$_{21}$ | .800 | .728 | .772 | .859 | .873 | .073 | .809 | .684 | .721 | .884 | .891 | .035 | .842 | .771 | .806 | .898 | .907 | .047 |
| C$^2$FNet$_{21}$ | .796 | .719 | .762 | .854 | .864 | .080 | .813 | .686 | .723 | .890 | .900 | .036 | .838 | .762 | .795 | .897 | .904 | .049 |
| UGTR$_{21}$ | .784 | .684 | .735 | .822 | .851 | .086 | .817 | .666 | .712 | .853 | .890 | .036 | .839 | .747 | .787 | .875 | .899 | .052 |
| PreyNet$_{22}$ | .790 | .708 | .757 | .842 | .857 | .077 | .813 | .697 | .736 | .881 | .891 | .034 | – | – | – | – | – | – |
| BSA-Net$_{22}$ | .794 | .717 | .763 | .851 | .867 | .079 | .818 | .699 | .738 | .891 | .901 | .034 | .841 | .771 | .808 | .897 | .907 | .048 |
| OCE-Net$_{22}$ | .802 | .723 | .766 | .852 | .865 | .080 | .827 | .707 | .741 | .894 | .905 | .033 | .853 | .785 | .818 | .903 | .913 | .045 |
| BGNet$_{22}$ | .812 | .749 | .789 | .870 | .882 | .073 | .831 | .722 | .753 | **.901** | .911 | .033 | .851 | .788 | .820 | .907 | .916 | .044 |
| SegMaR$_{22}$ | .815 | .753 | .795 | .874 | .884 | .071 | .833 | .724 | .757 | .899 | .906 | .034 | .841 | .781 | .820 | .896 | .907 | .046 |
| ZoomNet$_{22}$ | .820 | .752 | .794 | .878 | .892 | .066 | .838 | .729 | .766 | .888 | .911 | .029 | .853 | .784 | .818 | .896 | .912 | .043 |
| SINet-v2$_{22}$ | .820 | .743 | .782 | .882 | .895 | .070 | .815 | .680 | .718 | .887 | .906 | .037 | .847 | .770 | .805 | .903 | .914 | .048 |
| **Ours** | **.856** | **.799** | **.830** | **.899** | **.928** | **.050** | **.851** | **.735** | **.769** | .895 | **.930** | **.026** | **.879** | **.816** | **.843** | **.915** | **.937** | **.035** |

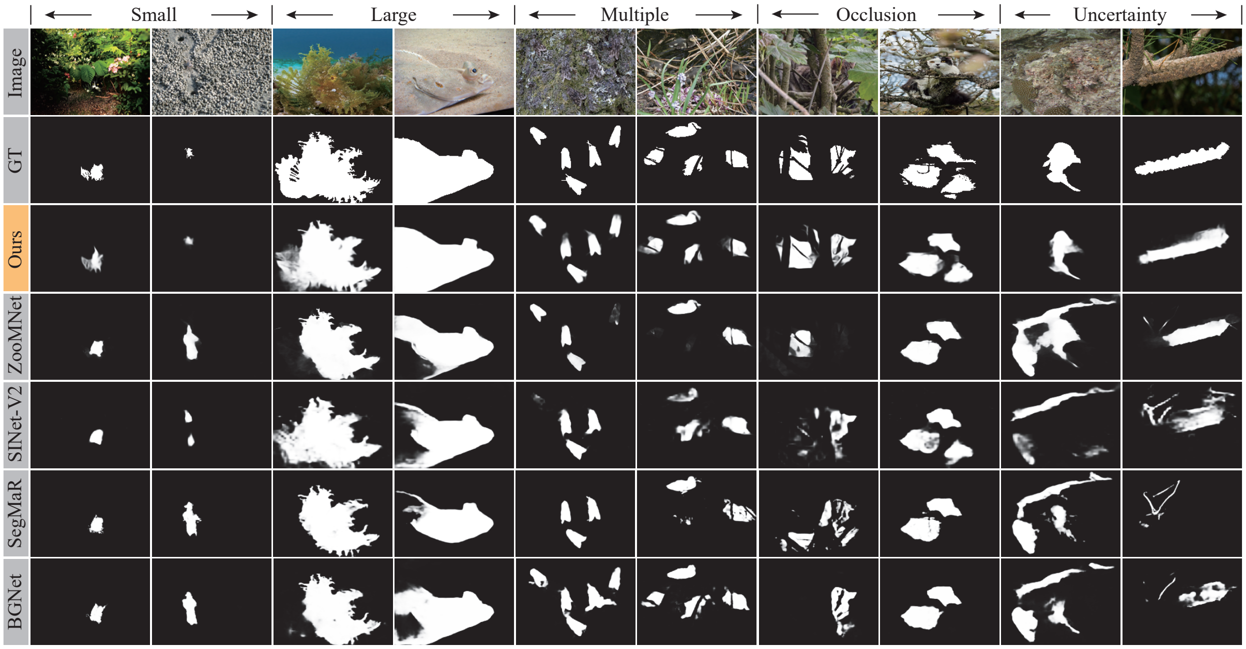

Figure 4. **Visual comparison with some representative SOTA models in challenging scenarios.** Please zoom in for details. More visual results are provided in the *Supplementary Material*.

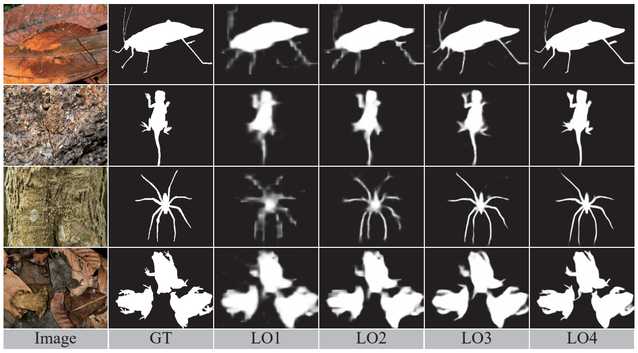

Figure 5. **Visual comparison of the lateral output of FSD**. From LO1 to LO4 (final output) denote the layers of FSD.

[27] ($F_\beta^\omega$), mean F-measure [1] ($F_\beta^m$), mean E-measure [9] ($E_\phi^m$), max E-measure ($E_\phi^x$), and mean absolute error ($\mathcal{M}$).

**Implementation Details.** The proposed model is implemented by PyTorch. The base version of ViT [7], pretrained by the DeiT strategy [40], is adopted as the transformer encoder. Other modules are randomly initialized. We follow the training set settings in [10,30] and adopt random flipping to augment the training data. All the input images are resized to 384×384. Adam is used as the optimizer, and the learning rate is initialized to 1e-4 and then scaled down by 10 every 50 epochs. The complete training process for 200 epochs with a batch size of 2 takes ~8 hours on a workstation with 8 NVIDIA Tesla V100 GPUs.

## 4.2. Comparison with State-of-the-Art Methods

To demonstrate the effectiveness of the proposed method, we compare it with 24 state-of-the-art methods, including 10 salient object detection methods (*i.e.*, BASNet [33], CPD-R [45], EGNet [54], SCRN [46], F$^3$Net [44], CSNet [13], SSAL [50], ITSD [58], UCNet [49], and VST [22]), and 14 COD methods (*i.e.*, SINet [11], SLSR [25], PFNet [28], MGL-R [48], UJSC [19], PreyNet [51], BSANet [59], C$^2$FNet [36], UGTR [47], OCE-Net [21], BGNet [37], SegMaR [16], ZoomNet [30], and SINet-v2 [10]). All the predictions of competitors are either provided by the authors or generated by models retrained based on the open-source codes. More experimental results are provided in the *Supplementary Material*.

**Quantitative Comparison.** Tab. 1 summarizes the quantitative results of our proposed method against 24 competitors on three challenging COD benchmark datasets under six evaluation metrics. It can be seen that the specially designed COD methods generally outperform the SOD models. Furthermore, our proposed method consistently surpasses all other models on these datasets. Compared to the recently proposed state-of-the-art ZoomNet [30], our method achieves average performance gains of 3.0%, 3.7%, 2.7%, 1.8%, 3.0%, and 17.7% in terms of $S_\alpha$, $F_\beta^w$, $F_\beta^m$, $E_\phi^m$, $E_\phi^x$, and $\mathcal{M}$ on these three datasets, respectively. Compared to the recently proposed SINet-v2 [10], the average gains are 4.2%, 7.2%, 6.0%, 1.4%, 3.0%, and 28.5%, respectively. Besides, compared to the transformer-based methods (*i.e.*, VST [22] and UGTR [47]), our method shows significant performance improvements of 7.8%, 16.3%, 13.2%, 6.2%, 5.7%, and 34.1% over VST and 6.0%, 12.1%, 9.3%, 6.3%, 5.9%, and 34.1% over UTGR on average for $S_\alpha$, $F_\beta^w$, $F_\beta^m$, $E_\phi^m$, $E_\phi^x$, and $\mathcal{M}$, respectively. The superiority in performance benefits from the compensation of the local feature modeling for the transformer backbones, and the smooth and progressive feature decoding to accumulate more subtle clues of the camouflage objects.

**Visual Comparison.** Fig. 4 shows the visual comparisons of our proposed method with some representative competitors in several typical scenarios, including small, large, multiple, occluded objects, and uncertain boundaries. It can be seen that the compared methods are prone to provide inaccurate object localization, incomplete object regions, or missing objects, resulting in inferior segmentation of camouflaged objects. Our proposed method shows superior visual performance for more accurate and complete predictions. Experiments also demonstrates the robustness of the proposed method to different challenging scenarios.

## 4.3. Ablation Study

To validate the effectiveness of the proposed modules for COD, we perform the following ablation studies on these COD benchmark datasets.

**Dense Integration Strategy.** Integrating multiple backbone features to improve prediction is widely used in segmentation tasks. Therefore, we test combinations of different lateral features of the base ViT (denoted as B) for decoding. The baseline decoder contains concatenation, reshape, and upsampling operations. The results are shown in the first four rows of Tab. 2, where $i$ in B$_i$ denotes the number of feature layers adopted for decoding. We can see that aggregating different feature layers benefit merging more clues, thereby improving the detection performance. In our experiments, aggregating all the transformer feature layers (*i.e.*, B$_{12}$) provided the best performance.

**Feature Shrinkage Decoder.** Tab. 2 (5th~7th) shows the results of our proposed decoder FSD (denoted as D) under different backbone feature combinations. Note that the number of pyramid layers is 2, 3, and 4 for "(B$_4$+D)", "(B$_8$+D)", and "(B$_{12}$+D)", respectively. We can see that FSD effectively improves the performance, showing the designed FSD well aggregates and retains critical features of different layers for accurate predictions. Moreover, Fig. 5 provides the outputs of different pyramidal decoder layers in FSD (from LO1 to LO4 by depth), validating the ability of FSD to recover object details and generate clear predictions gradually.

Besides, we conducted five experiments to verify the ef-

Table 2. Ablation studies of FSPNet on benchmark datasets. “B” is backbone, “D” is FSD and “T” is NL-TEM.

| **Settings** | **CAMO (250)** | | | | | | **COD10K (2,026)** | | | | | | **NC4K (4,121)** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_m\uparrow$ | $F_\beta^\omega\uparrow$ | $F_\beta^m\uparrow$ | $E_\phi^m\uparrow$ | $E_\phi^x\uparrow$ | $\mathcal{M}\downarrow$ | $S_m\uparrow$ | $F_\beta^\omega\uparrow$ | $F_\beta^m\uparrow$ | $E_\phi^m\uparrow$ | $E_\phi^x\uparrow$ | $\mathcal{M}\downarrow$ | $S_m\uparrow$ | $F_\beta^\omega\uparrow$ | $F_\beta^m\uparrow$ | $E_\phi^m\uparrow$ | $E_\phi^x\uparrow$ | $\mathcal{M}\downarrow$ |
| $\mathrm{B}_1$ | .774 | .685 | .732 | .813 | .839 | .089 | .791 | .659 | .709 | .855 | .882 | .042 | .828 | .747 | .795 | .881 | .901 | .051 |
| $\mathrm{B}_4$ | .781 | .693 | .738 | .818 | .845 | .086 | .801 | .668 | .713 | .861 | .889 | .040 | .835 | .755 | .802 | .885 | .909 | .049 |
| $\mathrm{B}_8$ | .795 | .715 | .746 | .827 | .856 | .081 | .807 | .679 | .725 | .867 | .897 | .039 | .841 | .766 | .807 | .887 | .913 | .048 |
| $\mathrm{B}_{12}$ | .798 | .726 | .755 | .837 | .868 | .079 | .812 | .697 | .732 | .871 | .901 | .038 | .853 | .781 | .813 | .901 | .918 | .046 |
| $\mathrm{B}_4+\mathrm{D}$ | .786 | .716 | .743 | .831 | .857 | .082 | .830 | .696 | .735 | .873 | .894 | .038 | .854 | .773 | .810 | .902 | .921 | .048 |
| $\mathrm{B}_8+\mathrm{D}$ | .807 | .731 | .759 | .839 | .871 | .076 | .831 | .711 | .740 | .882 | .912 | .036 | .866 | .787 | .823 | .905 | .922 | .043 |
| $\mathrm{B}_{12}+\mathrm{D}$ | .817 | .755 | .786 | .858 | .891 | .062 | .844 | .728 | .759 | .888 | .918 | .033 | .870 | .808 | .836 | .912 | .929 | .040 |
| $\mathrm{B}_4+\mathrm{D}+\mathrm{T}$ | .809 | .731 | .762 | .837 | .877 | .078 | .836 | .707 | .742 | .883 | .913 | .036 | .864 | .786 | .824 | .906 | .927 | .038 |
| $\mathrm{B}_8+\mathrm{D}+\mathrm{T}$ | .827 | .762 | .788 | .862 | .912 | .066 | .842 | .724 | .757 | .887 | .922 | .029 | .868 | .803 | .832 | .907 | .932 | .037 |
| $\mathbf{B_{12}+D+T}$ | **.856** | **.799** | **.830** | **.899** | **.928** | **.050** | **.851** | **.735** | **.769** | **.895** | **.930** | **.026** | **.879** | **.816** | **.843** | **.915** | **.937** | **.035** |

Table 3. More ablation studies on COD10K and NC4K.

| **No.** | **COD10K** | | | | | | **NC4K** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $S_m\uparrow$ | $F_\beta^\omega\uparrow$ | $F_\beta^m\uparrow$ | $E_\phi^m\uparrow$ | $E_\phi^x\uparrow$ | $\mathcal{M}\downarrow$ | $S_m\uparrow$ | $F_\beta^\omega\uparrow$ | $F_\beta^m\uparrow$ | $E_\phi^m\uparrow$ | $E_\phi^x\uparrow$ | $\mathcal{M}\downarrow$ |
| ① | .825 | .710 | .739 | .875 | .908 | .037 | .859 | .786 | .820 | .904 | .920 | .044 |
| ② | .848 | .731 | .764 | .891 | .923 | .027 | .875 | .811 | .837 | .910 | .924 | .037 |
| ③ | .840 | .722 | .753 | .882 | .916 | .034 | .867 | .798 | .832 | .906 | .921 | .039 |
| ④ | .849 | .732 | .761 | .887 | .922 | .029 | .872 | .804 | .832 | .901 | .927 | .038 |
| **Ours** | .851 | .735 | .769 | .895 | .930 | .026 | .879 | .816 | .843 | .915 | .937 | .035 |
| ⑤ | .844 | .728 | .759 | .888 | .918 | .033 | .870 | .808 | .836 | .912 | .929 | .040 |
| +GFM | .846 | .732 | .764 | .889 | .924 | .028 | .874 | .810 | .838 | .913 | .925 | .037 |
| +NL | .847 | .731 | .765 | .892 | .926 | .028 | .873 | .811 | .839 | .913 | .926 | .036 |

fectiveness of the decoder components and structures, including ① replacing FSD with an U-shaped decoding structure (similar to Fig. 2 (a)), ② replacing AIM with a simpler combination of operations (*i.e.*, concatenation and 1×1 convolution), ③ extending AIM to aggregate three adjacent feature layers, ④ adjusting our decoder to pairwise feature aggregation with overlap and removing lateral supervision and feature interaction within the same layer (similar to [26]). Note that we retain other modules in experiments. The results are shown in Tab. 3 (1st∼5th rows).

Our decoder and ④ outperforms the U-shaped decoding structure (①) by a large margin, this is because this type of decoder usually directly aggregates features (in the same fusion layer) with large feature differences, and tends to discard some subtle but valuable cues, resulting in inaccurate predictions, especially for the task of identifying camouflaged objects from faint clues. Our decoder and ④ both progressively aggregates adjacent features through a layer-by-layer shrinkage pyramid (multiple fusion layers) to accumulate valuable cues as much as possible for object prediction. However, our decoder introduces lateral supervision and feature flow within the same layer, which force the decoder to accumulate more critical camouflaged object cues, thus achieving a large performance improvement, especially on NC4K dataset, compared to ④. Besides, by comparing with ② and ③, the proposed AIM component provides better performance for camouflaged object prediction.

**Non-local Token Enhancement Module.** Tab. 2 (8th∼10th rows) shows the results of NL-TEM (denoted as $T$). The NL-TEM complements the local feature exploration for transformers, which contributes to the recovery of objects' local details, and further improves the performance.

Besides, we perform two additional experiments to verify the effectiveness of non-local operations and graph convolutions. The results are shown in Tab. 3 (6th∼8th rows). ⑤ denotes “$\mathrm{B}_{12}$+D”. Based on model ⑤, we add the GFM module (“+GFM”), that is, the two inputs of NL-TEM are directly fed into the GFM after reshape, concatenation and softmax operations. The “+NL” denotes removing the GFM directly from NL-TEM to test the non-local operations because the size of the input and output of the GFM are the same. It can be seen that the addition of non-local operations and GFM both contribute to camouflaged object detection and promote the improvement of detection performance. When combining these two components (*i.e.*, “$\mathrm{B}_{12}$+D+T” in Tab. 2), the proposed model significantly improves the performance for camouflaged object detection.

# 5. Conclusion

Considering the existing COD methods suffer from two issues, that is, less effective locality modeling for transformer-based models and limitations of feature aggregation in decoders, in this paper, we propose a novel transformer-based feature shrinkage pyramid network (FSPNet), which contains a non-local token enhancement module (NL-TEM) and a feature shrinkage decoder (FSD) with adjacent interaction modules (AIM). The proposed model can hierarchically aggregate locality-enhanced neighboring features through progressive shrinking, thereby integrating subtle but effective local and global cues as much as possible for accurate and complete camouflaged object detection. Extensive comparison experiments and ablation studies show that the proposed FSPNet achieves superior performance over 24 cutting-edge approaches on three widely-used COD benchmark datasets.

**Acknowledgments.** We thank Deng-Ping Fan for insightful feedback.

## References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009. 7

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Nagappa U Bhajantri and P Nagabhushan. Camouflage defect identification: a novel approach. In *International Conference on Information Technology (ICIT)*, pages 145–148, 2006. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 3

[5] Geng Chen, Si-Jie Liu, Yu-Jia Sun, Ge-Peng Ji, Ya-Feng Wu, and Tao Zhou. Camouflaged object detection via context-aware cross-level fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 1

[6] 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 *CVPR*, pages 13864–13873, 2022. 2

[7] 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 *ICLR*, 2021. 2, 3, 7

[8] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 5

[9] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, 2018. 7

[10] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 2022. 1, 2, 7

[11] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020. 1, 3, 5, 7

[12] 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 *MICCAI*, pages 263–273. Springer, 2020. 3

[13] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *ECCV*, pages 702–721. Springer, 2020. 7

[14] Xiaobin Hu, Deng-Ping Fan, Xuebin Qin, Hang Dai, Wenqi Ren, Ying Tai, Chengjie Wang, and Ling Shao. High-resolution iterative feedback network for camouflaged object detection. *arXiv preprint arXiv:2203.11624*, 2022. 3

[15] Zhou Huang, Huaixin Chen, Biyuan Liu, and Zhixi Wang. Semantic-guided attention refinement network for salient object detection in optical remote sensing images. *Remote Sensing*, 13(11):2163, 2021. 3

[16] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *CVPR*, pages 4713–4722, 2022. 2, 7

[17] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021. 2

[18] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 184:45–56, 2019. 2, 5

[19] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *CVPR*, pages 10071–10081, 2021. 2, 7

[20] Lin Li, Jingyi Liu, Shuo Wang, Xunkun Wang, and Tian-Zhu Xiang. Trichomonas vaginalis segmentation in microscope images. In *MICCAI*, pages 68–78. Springer Nature Switzerland, 2022. 1

[21] Jiawei Liu, Jing Zhang, and Nick Barnes. Modeling aleatoric uncertainty for camouflaged object detection. In *WACV*, pages 1445–1454, 2022. 2, 7

[22] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, pages 4722–4732, 2021. 7

[23] 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 *ICCV*, pages 10012–10022, 2021. 3

[24] Zhengyi Liu, Zhili Zhang, Wei Wu, et al. Boosting camouflaged object detection with dual-task interactive transformer. In *ICPR*, pages 1–7, 2022. 2, 3

[25] 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 *CVPR*, pages 11591–11601, 2021. 2, 5, 7

[26] Mingcan Ma, Changqun Xia, Jia Li, et al. Pyramidal feature shrinking for salient object detection. In *AAAI*, volume 35, pages 2311–2318, 2021. 5, 8

[27] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, pages 248–255, 2014. 7

[28] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, pages 8772–8781, 2021. 2, 7

[29] Melia G Nafus, Jennifer M Germano, Jeanette A Perry, Brian D Todd, Allyson Walsh, and Ronald R Swaisgood. Hiding in plain sight: a study on camouflage and habitat selection in a slow-moving desert herbivore. *Behavioral Ecology*, 26(5):1389–1394, 2015. 1

[30] 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 *CVPR*, 2022. 1, 2, 3, 7

[31] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. Multi-scale interactive network for salient object detection. In *CVPR*, pages 9413–9422, 2020. 3

[32] Jialun Pei, Tianyang Cheng, Deng-Ping Fan, He Tang, Chuanbo Chen, and Luc Van Gool. Osformer: One-stage camouflaged instance segmentation with transformers. In *ECCV*, 2022. 3

[33] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, pages 7479–7489, 2019. 7

[34] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *CVPR*, 2022. 2

[35] Avishek Siris, Jianbo Jiao, Gary KL Tam, Xianghua Xie, and Rynson WH Lau. Scene context-aware salient object detection. In *ICCV*, pages 4156–4166, 2021. 3

[36] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. In *IJCAI*, pages 1025–1031, 2021. 2, 7

[37] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection. In *IJCAI*, 2022. 2, 7

[38] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *ICML*, pages 10183–10192. PMLR, 2021. 3

[39] Gusi Te, Yinglu Liu, Wei Hu, Hailin Shi, and Tao Mei. Edge-aware graph representation learning and reasoning for face parsing. In *ECCV*, pages 258–274. Springer, 2020. 4, 5

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 3, 7

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NIPS*, 30, 2017. 3

[42] 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 *ICCV*, pages 568–578, 2021. 3

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 3, 4

[44] Jun Wei, Shuhui Wang, and Qingming Huang. $F^3$net: fusion, feedback and focus for salient object detection. In *AAAI*, pages 12321–12328, 2020. 7

[45] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. In *CVPR*, pages 3907–3916, 2019. 7

[46] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *ICCV*, pages 7264–7273, 2019. 7

[47] 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 *ICCV*, pages 4146–4155, 2021. 2, 3, 7

[48] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *CVPR*, pages 12997–13007, 2021. 2, 3, 7

[49] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Tong Zhang, and Nick Barnes. Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In *CVPR*, pages 8582–8591, 2020. 7

[50] Jing Zhang, Xin Yu, Aixuan Li, Peipei Song, Bowen Liu, and Yuchao Dai. Weakly-supervised salient object detection via scribble annotations. In *CVPR*, pages 12546–12555, 2020. 7

[51] Miao Zhang, Shuang Xu, Yongri Piao, Dongxiang Shi, Shusen Lin, and Huchuan Lu. Preynet: Preying on camouflaged objects. In *ACM MM*, pages 5323–5332, 2022. 7

[52] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *ICCV*, pages 202–211, 2017. 3

[53] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, pages 4320–4328, 2018. 3

[54] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, pages 8779–8788, 2019. 7

[55] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51. Springer, 2020. 3

[56] 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 *CVPR*, pages 6881–6890, 2021. 3

[57] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *CVPR*, pages 4504–4513, 2022. 2

[58] Huajun Zhou, Xiaohua Xie, Jian-Huang Lai, Zixuan Chen, and Lingxiao Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *CVPR*, pages 9141–9150, 2020. 3, 7

[59] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jing Qin. I can find you! boundary-guided separated attention network for camouflaged object detection. In *AAAI*, 2022. 2, 3, 7

[60] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu. Inferring camouflaged objects by texture-aware interactive guidance network. In *AAAI*, pages 3599–3607, 2021. 2