Boosting Foreground-Background Disentanglement for Camouflaged Object Detection

JIESHENG WU, School of Computer and Information, Anhui Normal University, China
FANGWEI HAO, College of Artificial Intelligence, Nankai University, China
JING XU*, College of Artificial Intelligence, Nankai University, China

In nature, certain objects exhibit patterns that closely resemble their backgrounds, a phenomenon commonly referred to as Camouflaged Object Detection (COD). We argue that existing COD approaches often suffer from insufficient discriminability for these objects, which we attribute to a lack of effective disentangling of foreground and background representations. To address this, we propose a novel Foreground-Background Disentanglement Network (FBD-Net) that enhances foreground-background disentanglement learning to improve discriminability. Specifically, we design an Edge-guided Foreground-Background Decoupling (EFBD) module, which facilitates the separated learning of foreground and background representations. Additionally, we introduce the Foreground-Background Representation Disentangling Head (DisHead) to further boost the discriminative power of the model. The DisHead consists of two objectives: the Edge Objective and the FoBa Objective. Furthermore, we propose three complementary modules: the Context Aggregation Module (CAM) for initial coarse object detection, the Scale-Interaction Enhanced Pyramid (SIEP) for multi-scale information extraction, and the Cross-Stage Adaptive Fusion (CSAF) module for subtle clue accumulation. Extensive experiments demonstrate that both our CNN-based and Transformer-based FBD-Nets outperform 26 state-of-the-art COD methods across four public datasets. Codes will be released on https://github.com/TomorrowJW/FBD-Net-COD.

CCS Concepts: • **Computing methodologies** → **Computer vision tasks**.

Additional Key Words and Phrases: Camouflaged object detection, Disentanglement learning, Edge-guided, Contrastive learning

1 INTRODUCTION

Camouflaged Object Detection (COD) [13] aims to accurately localize and segment objects that blend into their surrounding environments. This task is notably more challenging than standard object or salient object detection due to the intrinsic high visual similarity between camouflaged objects and the background. Such similarity imposes higher demands on a model's **discriminative ability**, requiring it to capture **subtle and fine-grained differences** in texture, boundary, and semantics to succeed. COD has found increasing application in diverse fields, including species discovery [50], medical imaging (e.g., polyp segmentation [14]), industrial defect detection [3], military surveillance [82], and autonomous driving [57]. In recent years, most COD models have relied on deep learning. Existing methods can be broadly categorized into three paradigms: (1) designing task-specific modules for camouflage perception [5, 12, 22, 84], (2) incorporating prior knowledge such as edges and frequency

*Corresponding author

Authors' Contact Information: Jiesheng Wu, jasonwu@mail.nankai.edu.cn, School of Computer and Information, Anhui Normal University, Wuhu, China; Fangwei Hao, College of Artificial Intelligence, Nankai University, Tianjin, China, haofangwei@mail.nankai.edu.cn; Jing Xu, College of Artificial Intelligence, Nankai University, Tianjin, China, xujing@nankai.edu.cn.

ACM 1551-6865/2025/9-ART
https://doi.org/10.1145/3768584

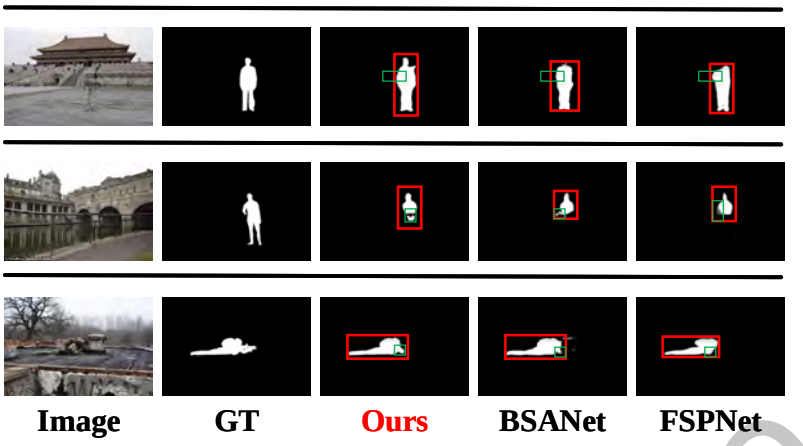


Fig. 1. Several visual prediction maps of camouflaged scene images are presented to compare our methods with recent SOTA methods (*e.g.,* BSANet [86], FSPNet [22]). Compared to these two methods, our approach offers greater advantages in preserving object ***structure integrity*** (marked by red boxes) and capturing ***finer details*** (marked by green boxes).

cues [33, 35, 67, 76, 77, 83], and (3) leveraging auxiliary tasks (e.g., uncertainty prediction, object counting) to promote collaborative learning [32, 72]. Despite their progress, these approaches commonly train models using only **cross-entropy loss**, which implicitly treats foreground and background as a unified representation. As a result, the learned features are **entangled**, limiting the model's ability to distinguish camouflaged targets, particularly in complex scenes.

As illustrated in Fig. 1, even state-of-the-art models such as BSANet [86] and FSPNet [22] fail to preserve the **structure integrity** and **fine details** of camouflaged targets. We attribute this to their **insufficient foreground-background disentanglement**, which undermines their discriminative capacity.

To address this limitation, we propose a novel **Foreground-Background Disentanglement Network (FBD-Net)**. Unlike prior approaches that rely on implicit cues, our method introduces a **disentanglement learning strategy** that explicitly separates foreground and background representations at both the feature and supervision levels. This strategy is realized through two core modules: The **Edge-guided Foreground-Background Decoupling (EFBD)** module decomposes features into two distinct edge-guided flows: a *foreground flow* to capture object-related semantics and a *background flow* to model environmental cues. Edge information is injected into both to enhance the boundary perception of camouflaged objects. The **Foreground-Background Representation Disentangling Head (DisHead)** further promotes feature separation through two objectives: an *Edge Objective* using element-wise subtraction for edge prediction, and a *FoBa (Foreground-Background) Objective* based on contrastive learning to maximize the distance between global foreground and background embeddings.

These two modules explicitly drive the network to learn separable, discriminative feature representations for foreground and background, thereby improving both object localization and segmentation. In addition to the core disentangling design, we incorporate three auxiliary modules to support efficient camouflage understanding: The **Context Aggregation Module (CAM)** performs coarse object localization. The **Scale-Interaction Enhanced Pyramid (SIEP)** extracts multi-scale contextual information. The **Cross-Stage Adaptive Fusion (CSAF)** module accumulates subtle camouflage clues across layers.

To summarize, our key contributions are as follows:

- We propose FBD-Net, a novel framework that introduces explicit foreground-background disentanglement to boost model discriminability in COD.
- We design the EFBD and DisHead modules to learn fine-grained, separable representations of foreground and background, guided by edge-aware and contrastive supervision.
- We incorporate CAM, SIEP, and CSAF modules to further improve camouflage recognition through contextual aggregation and cross-scale fusion.
- Extensive experiments on four public benchmarks show that our method outperforms 26 state-of-the-art COD models using both CNN and Transformer backbones.

2 RELATED WORK

In this section, we will review three types of research related to our work, including camouflaged object detection, disentangled learning, and contrastive learning.

2.1 Camouflaged Object Detection

Camouflaged Object Detection (COD) is the task of segmenting objects that closely resemble their surroundings. Recently, the majority of researchers have turned to deep learning-based methods to address COD challenges. Specifically, Fan *et al.* [13] introduced the COD10K dataset, a large-scale resource aimed at advancing COD research. Their pioneering work included the development of a bio-inspired learning paradigm, designing a search module, and an identification module tailored for COD. The paradigm has served as a valuable source of inspiration for subsequent research. Its core concept involves devising a context module for preliminary object localization and a decoder to progressively refine the segmentation. For instance, Zhang *et al.* [78] proposed PreyNet, which employed sensory and predator mechanisms for COD. Liu *et al.* [37] built upon this idea with MSCAF-Net, integrating multi-scale context features. Other notable works inspired by this approach include ERRNet [25], PFNet [42], C2FNet [5], FAPNet [84], SINetV2 [12], and more. In our own methodology, we also leverage this concept, designing a context module known as CAM.

Another promising avenue in improving COD performance is the incorporation of prior knowledge and auxiliary tasks. Drawing inspiration from the role of edge guidance in Salient Object Detection (SOD) tasks, several noteworthy approaches have integrated edge information into COD. Zhu *et al.* [86] proposed BSANet that used two-stream separated attention modules to detect edge. Sun *et al.* [59] aggregated low-level and high-level features for edge learning. Moreover, Zhong *et al.* [83] and Li *et al.* [33] efficiently injected frequency knowledge for COD performance improvement. As for auxiliary tasks, Li *et al.* [32] pioneered the introduction of a joined framework to achieve adversarial learning between SOD and COD tasks, which improved performance. Similar work also included multi-task learning network [72] and texton-coherence network [77]. Beyond these, innovative works such as LSR [40], MGL [76], Zoom-Net [46], SegMaR [26], FSPNet [22], and CamoFormer [73]. Recently, some excellent works such as DCNet [75], DRFNet [63], PRNet [21], PRBE-Net [74], and SENet [17] apply novel learning strategies to enhance COD or weakly supervised COD [30, 44].

The exemplary works mentioned above have greatly informed our research, and some of their concepts have influenced our approach. Nevertheless, in contrast to these studies, our primary objective is to empower the model to deepen its discriminability. As a result, our entire methodology revolves around the concept of disentangled learning, enabling the model to discern the distinction between the camouflaged object and the background effectively.

2.2 Disentangled Learning and Contrastive Learning

Disentangled Learning (DL) refers to the process of embedding task-related and irrelevant concepts and semantics into distinct feature dimensions rather than solely extracting the most relevant information for the task [1, 8, 19].

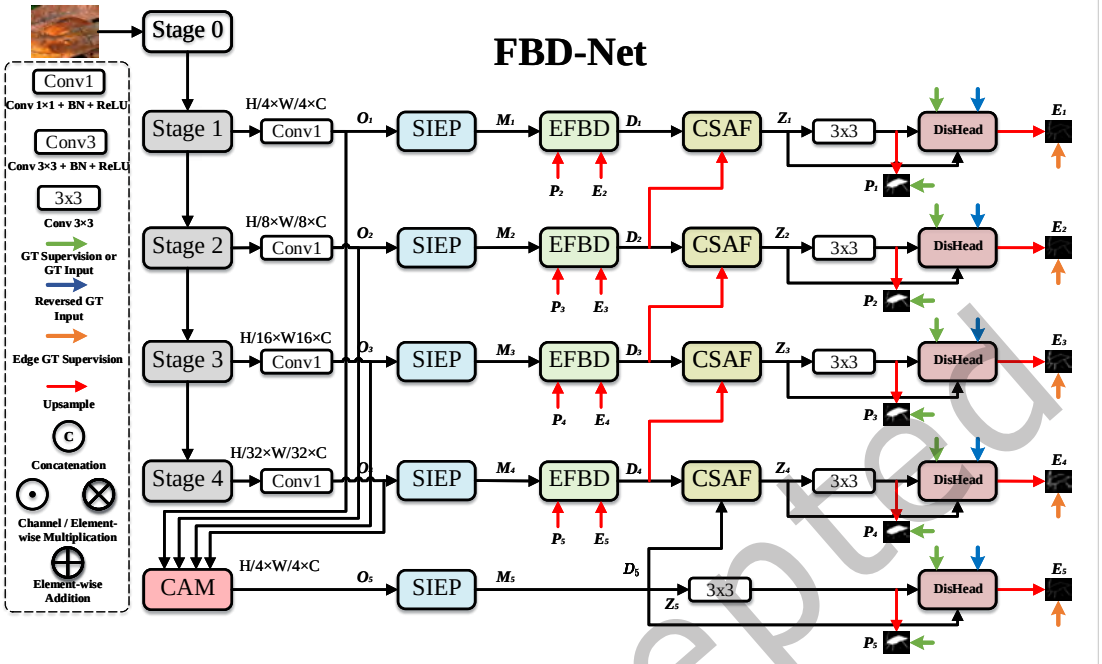


Fig. 2. Overall architecture of the proposed FBD-Net. FBD-Net consists of five key components: a Context Aggregation Module (CAM), a Scale-Interaction Enhance Pyramid (SIEP), an Edge-guided Foreground-Background Decoupling (EFBD) module, a Cross-Stage Adaptive Fusion (CSAF) module, and a Foreground-Background Representation Disentangling Head (DisHead).

The methodology is frequently utilized in representation learning and architecture design. For instance, the renowned recent work, RevCol [4], introduced a reversible columnar network based on this principle, enabling information to flow through the network without any losses. This network significantly enhanced the performance of a wide range of downstream tasks. In our work, we borrow this concept and develop an EFBD module that decouples foreground and background features and learns them separately. Additionally, we propose a DisHead to enhance the achievements of disentangled learning.

Contrastive Learning (CL) is frequently employed in discriminative self-supervised pre-training tasks. Its central idea is to construct pairs of positive and negative samples so that models can better comprehend the relationship and discernibility between samples with the aid of InfoNCE loss [45]. MoCo [18] and SimCLR [6] are two exemplary works that utilize contrastive learning to achieve outstanding results. The former primarily addresses the consistency issues in previous contrastive learning methods, while the latter adopts a simplified framework to implement contrastive learning but relies on a larger batch size. Furthermore, supervised contrastive learning [27] extends upon contrastive learning and achieves remarkable success. In our work, we introduce a novel contrastive learning paradigm for foreground-to-background contrast to enhance representation disentanglement.

3 PROPOSED METHOD

3.1 Overview

The overall architecture of FBD-Net is illustrated in Fig. 2. FBD-Net consists of five key components: a Context Aggregation Module (CAM), a Scale-Interaction Enhance Pyramid (SIEP), an Edge-guided Foreground-Background

Decoupling (EFBD) module, a Cross-Stage Adaptive Fusion (CSAF) module, and a Foreground-Background Representation Disentangling Head (DisHead). We take CNN-based and Transformer-based backbones as our encoders, all of which have a similar deep architecture of five stages. Moreover, our decoder decodes predictions layer by layer in a top-down manner. Here, we take Res2Net-50 [16] as an example encoder to introduce our FBD-Net. Specifically, an image is first fed into the encoder to extract the top four stage features $\{\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \boldsymbol{O}_4\}$, respectively, which are followed with four 1×1 convolutions to reduce channels. Then, inspired by the previous works [12, 37], to achieve semantic consistency during decoding and obtain an initial coarse prediction for subsequent decoding, we feed the extracted features into our proposed CAM to generate comprehensive features $\boldsymbol{O}_5$. After that, these features are fed into SIEP and EFBD modules to boost camouflaged object representations. Moreover, to fuse subtle context clues, a CSAF module is designed to fuse cross-stage semantics for the following predictions. Finally, DisHeads are used to push the model to distinguish camouflaged objects and backgrounds as much as possible, which further boosts the camouflage representation understanding and discriminative ability of FBD-Net.

3.2 Context Aggregation Module

As discussed above, we follow the previous design paradigm [12, 37], which first aggregates multi-stage features to generate coarse context features for initial prediction. However, they directly modify the Partial Decoder Component (PDC) in [68], integrating these multi-level features via element-wise multiplication operations. Such operations may lead to feature confusion and error accumulation, which is not conducive to the subsequent learning of subtle features. Unlike their methods, we not only consider bridging context semantics but also maintain the independent characteristics of these features. Thus, we employ multiple 3×3 convolutional layers and concatenation operations to obtain contextual features. The details of CAM are shown in Fig. 3. More

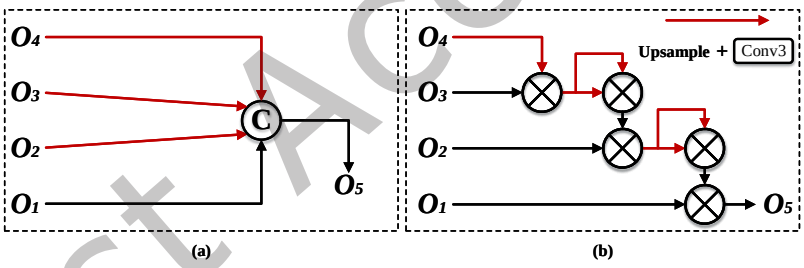


Fig. 3. (a) Details of the proposed Context Aggregation Module (CAM). (b) The modified PDC structure in [12, 37].

specifically, given the inputs $\{\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \boldsymbol{O}_4\}$, the context features $\boldsymbol{O}_5$ are obtained as

$$\begin{cases} \boldsymbol{O}_4 = \delta_{\uparrow}\left(\mathrm{Conv}_3\left(\boldsymbol{O}_4\right)\right), \\ \boldsymbol{O}_3 = \delta_{\uparrow}\left(\mathrm{Conv}_3\left(\boldsymbol{O}_3\right)\right), \\ \boldsymbol{O}_2 = \delta_{\uparrow}\left(\mathrm{Conv}_3\left(\boldsymbol{O}_2\right)\right), \\ \boldsymbol{O}_1 = \mathrm{Conv}_3\left(\boldsymbol{O}_1\right), \\ \boldsymbol{O}_5 = \mathrm{Conv}_3\left(\left[\boldsymbol{O}_1, \boldsymbol{O}_2, \boldsymbol{O}_3, \boldsymbol{O}_4\right]\right), \end{cases} \tag{1}$$

where $\mathrm{Conv}_3(\cdot)$ denotes a 3×3 convolutional layer followed by a batch normalization and a ReLU activation (**Abbreviated as CBR component**). $\delta_{\uparrow}(\cdot)$ denotes the upsampling operation. $[\cdots]$ represents the concatenation operation. $\boldsymbol{O}_5$ is used to decode coarse prediction and participate in the subsequent decoding processes.

3.3 Scale-Interaction Enhance Pyramid

Feature pyramids are widely-used strategies in COD [12, 33, 46, 59, 78, 84], which aim to capture multi-scale semantics for boosting camouflaged object representations. However, most existing methods usually independently perform multi-scale extraction without inter-scale interactions. An intuitive idea is that features on each scale should interact with and be promoted by each other. Moreover, for COD tasks, since there are similar patterns between objects and background, it is crucial to accumulate subtle but effective camouflaged cues as much as possible in the forward process of the feature flows [22]. Motivated by these insights, we propose the SIEP fuse multi-scale features better. SIEP incorporates multiple residual connections and designs camouflaged representation enhancement (CRE) components. The former is used for inter-scale interactions, and the latter is employed to boost camouflaged cue mining.

The structure of SIEP is shown in Fig. 4. Concretely, given the input features $\boldsymbol{O}_i$, we first feed $\boldsymbol{O}_i$ into four 1×1 CBR components to obtain four independent branches as

$$\boldsymbol{O}_i^j = \mathrm{Conv}_1\left(\boldsymbol{O}_i\right), j \in \{1, 2, 3, 4\}, \tag{2}$$

where each branch is used to capture uni-scale features. Then, each branch is fed into two successive components:

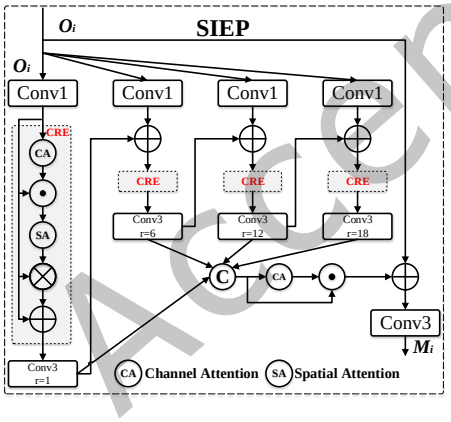


Fig. 4. Overall structure of the proposed Scale-Interaction Enhance Pyramid (SIEP) module.

the CRE component and dilated convolution. For the CRE components, we employ typical channel and spatial attention [65] to capture subtle but effective camouflaged cues as much as possible. Unlike the CBAM component [65], our CRE component multiplies the original features $\boldsymbol{O}_i^j$ with the features after channel and spatial attention throughout the process, rather than multiplying the features after channel attention with those after spatial attention. Additionally, we introduce a residual connection to continuously refine and purify the camouflage clues embedded in the original features. For the dilated convolutions, we use four 3×3 CBR components with dilated rates $r = \{1, 6, 12, 18\}$. The entire forward process can be formulated as

$$\boldsymbol{O}_i^{j_r} = \mathrm{Conv}_3^{j_r}\underbrace{\left(\boldsymbol{O}_i^j + \boldsymbol{O}_i^j \otimes \mathrm{SA}\left(\boldsymbol{O}_i^j \odot \mathrm{CA}\left(\boldsymbol{O}_i^j\right)\right)\right)}_{\mathrm{CRE}}, \tag{3}$$

where $\mathrm{Conv}_3^{j_r}(\cdot)$ denotes the 3×3 CBR component with a dilated rate of r. $\mathrm{CA}(\cdot)$ and $\mathrm{SA}(\cdot)$ are the channel and spatial attention, respectively. $\odot$ and $\otimes$ denote the channel-wise and element-wise multiplications, respectively.

Next, we conduct multi-scale interaction learning, which can be formulated as

$$\begin{cases} \boldsymbol{O}_i^{j_r} = \mathrm{Conv}_3^{j_r}\left(\mathrm{CRE}\left(\boldsymbol{O}_i^{j}\right)\right), & j = 1, \\ \boldsymbol{O}_i^{j_r} = \mathrm{Conv}_3^{j_r}\left(\mathrm{CRE}\left(\boldsymbol{O}_i^{j_r-1} + \boldsymbol{O}_i^{j}\right)\right), & j = 2, 3, 4. \end{cases} \tag{4}$$

Finally, all obtained multi-scale features are concatenated, and a residual connection, channel attention, and a 3×3 CBR component is employed to obtain the final output $\boldsymbol{M}_i$. $\boldsymbol{M}_i$ can be written as

$$\boldsymbol{M}_i = \mathrm{Conv}_3(\boldsymbol{O}_c \odot \mathrm{CA}(\underbrace{[\boldsymbol{O}_i^1, \boldsymbol{O}_i^2, \boldsymbol{O}_i^3, \boldsymbol{O}_i^4]}_{\boldsymbol{O}_c})) + \boldsymbol{O}_i). \tag{5}$$

3.4 Edge-guided Foreground-Background Decoupling Module

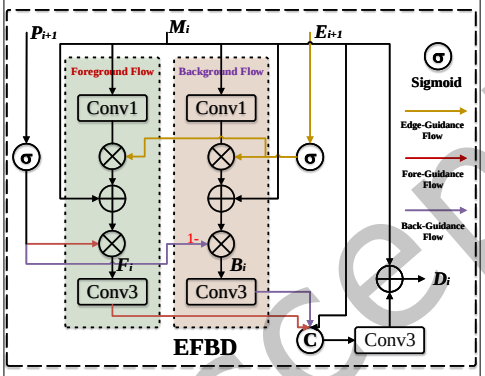


Fig. 5. Overall structure of the proposed Edge-guided Foreground-Background Decoupling (EFBD) module.

Camouflaged objects typically exhibit similar patterns to their surroundings, thus confusing the eyes of hunters and making them difficult to detect. However, during the learning process of a model, due to the inherent properties of the camouflage scene, the learning of foreground objects and background is usually entangled, which may not be conducive to the model learning camouflage representation. Therefore, we propose a novel separated decoupling learning paradigm. To be specific, for the obtained multi-scale features $\boldsymbol{M}_i$, we use the EFBD module to decouple the foreground and background representations as much as possible, which encourages the model to focus its attention on imperceptible camouflaged objects. Such a separated learning paradigm can boost the understanding of camouflaged objects.

Specifically, we use the prediction map $\boldsymbol{P}_i$ and edge map $\boldsymbol{E}_i$ predicted in the previous stage to construct our EFBD module, whose structure is detailed in Fig. 5. Given the $\boldsymbol{M}_i$ in the current stage and $\boldsymbol{P}_{i+1}$ and $\boldsymbol{E}_{i+1}$ in the previous stage, we design two separate flows to guide the model to achieve separated learning. First, for the foreground flow, we can obtain foreground representation $\boldsymbol{F}_i$ as

$$\boldsymbol{F}_i = \mathcal{S}\left(\boldsymbol{P}_{i+1}\right) \otimes \left(\boldsymbol{M}_i + \mathrm{Conv}_1\left(\boldsymbol{M}_i\right) \otimes \mathcal{S}\left(\boldsymbol{E}_{i+1}\right)\right), \tag{6}$$

where $\mathcal{S}(\cdot)$ denotes the Sigmoid activation.

Similarity, for the background flow, the background representation $\boldsymbol{B}_i$ can be formulated as

$$\boldsymbol{B}_i = \left(\mathbf{1} - \mathcal{S}\left(\boldsymbol{P}_{i+1}\right)\right) \otimes \left(\boldsymbol{M}_i + \mathrm{Conv}_1\left(\boldsymbol{M}_i\right) \otimes \mathcal{S}\left(\boldsymbol{E}_{i+1}\right)\right). \tag{7}$$

Then, $\boldsymbol{F}_i$ and $\boldsymbol{B}_i$ are fed into two independent 3×3 CBR components to enhance foreground and background feature representations, respectively. Finally, $\boldsymbol{F}_i$, $\boldsymbol{B}_i$, and $\boldsymbol{M}_i$ are concatenated, and followed by a 3×3 CBR

component and a residual connection to output the discriminative features $\boldsymbol{D}_i$ ($i \in \{1, 2, 3, 4\}$). The process is formulated as

$$\boldsymbol{D}_i = \boldsymbol{M}_i + \mathrm{Conv}_3\left(\left[\mathrm{Conv}_3\left(\boldsymbol{F}_i\right), \mathrm{Conv}_3\left(\boldsymbol{B}_i\right), \boldsymbol{M}_i\right]\right). \tag{8}$$

3.5 Cross-Stage Adaptive Fusion Module

Although the discriminative features $\boldsymbol{D}_i$ are obtained, accumulating and excavating subtle details and semantics across different stages are essential for the decoder. Moreover, we apply a top-down manner to decode the predictions, and each stage contains different valuable information for COD. Hence, a natural idea that considers a selective cross-stage feature strategy is motivated. Based on this, we employ an effective cross-stage fusion strategy to decode the desired outputs and inspired by SKNet [34], we propose a CSAF module. The details of the CSAF module are shown in Fig. 6.

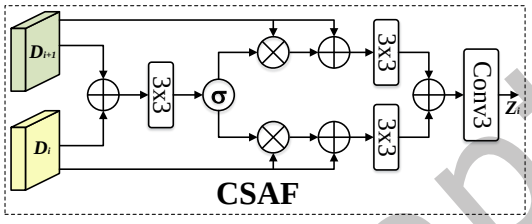


Fig. 6. Details of the proposed Cross-Stage Adaptive Fusion module (CSAF).

Specifically, we take the $\boldsymbol{D}_i$ and $\boldsymbol{D}_{i+1}$ as the inputs of the CSAF module (Note that for the fourth stage, we use $\boldsymbol{M}_5$ as $\boldsymbol{D}_5$). First, $\boldsymbol{D}_i$ and $\boldsymbol{D}_{i+1}$ are fused by an element-wise addition. Then, the fused features are fed into a 3×3 convolutional layer followed by a Sigmoid activation, aiming to generate weight maps to measure $\boldsymbol{D}_i$ and $\boldsymbol{D}_{i+1}$. The manner can adaptively fuse two features to enhance camouflaged features further. Next, two independent residual connections are implemented to preserve the initial semantics of input features. Finally, two features are fed into separated 3×3 convolutional layers, and further, an element-wise addition operation is performed followed by a 3×3 CBR component to boost feature representations. Therefore, the entire forward process can be denoted as

$$\begin{cases} \boldsymbol{W}_i = \mathcal{S}\left(\mathrm{Conv}_{3\times3}\left(\boldsymbol{D}_i + \boldsymbol{D}_{i+1}\right)\right), \\ \boldsymbol{H}_i = \mathrm{Conv}_{3\times3}\left(\boldsymbol{W}_i \otimes \boldsymbol{D}_i + \boldsymbol{D}_i\right), \\ \boldsymbol{H}_{i+1} = \mathrm{Conv}_{3\times3}\left(\boldsymbol{W}_i \otimes \boldsymbol{D}_{i+1} + \boldsymbol{D}_{i+1}\right), \\ \boldsymbol{Z}_i = \mathrm{Conv}_3\left(\boldsymbol{H}_i + \boldsymbol{H}_{i+1}\right), \end{cases} \tag{9}$$

where $\mathrm{Conv}_{3\times3}(\cdot)$ denotes a pure 3×3 convolutional layer. $\boldsymbol{W}_i$ represents the normalized weight maps, and $\boldsymbol{Z}_i$ ($i \in \{1, 2, 3, 4\}$) denotes the final outputs. $\boldsymbol{H}_i$ and $\boldsymbol{H}_{i+1}$ are the intermediate variables.

3.6 Foreground-Background Representation Disentangling Head

So far, we have obtained the learned feature representations $\boldsymbol{Z}_i$ from the decoder, and $\boldsymbol{Z}_i$ contains desired camouflage features. Thus, $\boldsymbol{Z}_i$ is fed into a 3×3 convolutional layer to output the predicted map $\boldsymbol{P}_i$ ($i \in \{1, 2, 3, 4, 5\}$). Note that for the fifth stage, we use $\boldsymbol{M}_5$ as $\boldsymbol{Z}_5$. Next, $\boldsymbol{P}_i$ is supervised by Ground Truth (GT) and uses Cross-Entropy (CE) loss to complete the training process. However, the CE Loss can only assign a fixed label to each pixel, which may not explicitly encourage the creation of large margins [9, 36, 58, 81] between the foreground and background representations. Additionally, a well-structured embedding space is crucial for COD, as it

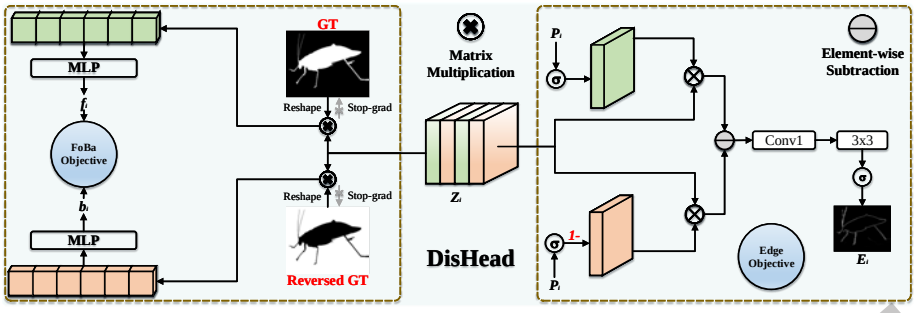


Fig. 7. Details of the proposed Foreground-Background Representation Disentangling Head (DisHead).

should be highly discriminative to distinguish the objects from their surroundings and fully learn camouflage representation. Based on this insight, we propose the Foreground-Background Representation Disentangling Head (DisHead) to encourage the entire model to distinguish camouflaged objects and backgrounds, further pushing the discriminative ability and camouflage object understanding of FBD-Net. The details of the DisHead are shown in Fig. 7.

As shown in Fig. 2 and Fig. 7, DisHead takes GT $\boldsymbol{G} \in \mathbb{R}^{h\times w\times 1}$, reversed GT $\boldsymbol{G}_R \in \mathbb{R}^{h\times w\times 1}$, predicted map $\boldsymbol{P}_i \in \mathbb{R}^{h\times w\times 1}$, and $\boldsymbol{Z}_i \in \mathbb{R}^{h\times w\times c}$ as its inputs, where h and w are the height and width while c is the number of channel. DisHead consists of two disentangling strategies: FoBa objective (left side of Fig. 7) and edge objective (right side of Fig. 7). First, since the edges between camouflaged objects and backgrounds are usually not sharp [84], we propose an edge objective to promote the model to locate the camouflage edge. Second, with the help of $\boldsymbol{G}$ and $\boldsymbol{G}_R$, we can obtain the global foreground embedding $\boldsymbol{f}_i \in \mathbb{R}^c$ and global background embedding $\boldsymbol{b}_i \in \mathbb{R}^c$, and they can be pushed away as much as possible by our proposed FoBa objective. Next, we describe the specific details of these two objectives.

3.6.1 Edge Objective. Given the $\boldsymbol{Z}_i$ and $\boldsymbol{P}_i$, we first conduct a Sigmoid activation on $\boldsymbol{P}_i$ to obtain foreground and background masks. Then, the obtained two masks are element-wise multiplied by $\boldsymbol{Z}_i$ respectively. Next, we employ an element-wise subtraction between the weighted two tensors to output the desired edge map $\boldsymbol{E}_i$:

$$\begin{cases} \boldsymbol{E}_t = \left(\mathcal{S}\left(\boldsymbol{P}_i\right) \otimes \boldsymbol{Z}_i\right) \ominus \left(\left(\mathbf{1} - \mathcal{S}\left(\boldsymbol{P}_i\right)\right) \otimes \boldsymbol{Z}_i\right), \\ \boldsymbol{E}_i = \mathrm{Conv}_{3\times 3}\left(\mathrm{Conv}_1\left(\boldsymbol{E}_t\right)\right), \end{cases} \tag{10}$$

where $\ominus$ denotes the element-wise subtraction operation. $\boldsymbol{E}_i$ is supervised by the edge GT ($\boldsymbol{G}_E$).

3.6.2 FoBa Objective. FoBa Objective aims to promote the model to distinguish camouflaged objects and backgrounds as much as possible, which further boost the discriminative ability of FBD-Net. As shown in the left side of Fig. 7, given $\boldsymbol{G} \in \mathbb{R}^{h\times w\times 1}$, $\boldsymbol{G}_R \in \mathbb{R}^{h\times w\times 1}$, and $\boldsymbol{Z}_i \in \mathbb{R}^{h\times w\times c}$ as its inputs, they first are reshaped as $\boldsymbol{G} \in \mathbb{R}^{1\times hw}$, $\boldsymbol{G}_R \in \mathbb{R}^{1\times hw}$, and $\boldsymbol{Z}_i \in \mathbb{R}^{hw\times c}$. Then, we use matrix multiplications and two two-layer MLP with non-linear activations to output $\boldsymbol{f}_i \in \mathbb{R}^c$ and $\boldsymbol{b}_i \in \mathbb{R}^c$. This can be formulated as

$$\begin{cases} \boldsymbol{f}_i = \mathrm{MLP}\left(\boldsymbol{G} \times \boldsymbol{Z}_i\right), \\ \boldsymbol{b}_i = \mathrm{MLP}\left(\boldsymbol{G}_R \times \boldsymbol{Z}_i\right), \end{cases} \tag{11}$$

where $\times$ denotes the matrix multiplication.

Now, we have obtained the global foreground and background embeddings $\boldsymbol{f}_i$ and $\boldsymbol{b}_i$. Next, we will use FoBa objective to push them away. Specifically, we use contrastive learning [6, 18, 45] to achieve the objective. However, **how to determine the positive pairs and negative pairs?** The specific structure is shown in Fig. 8.

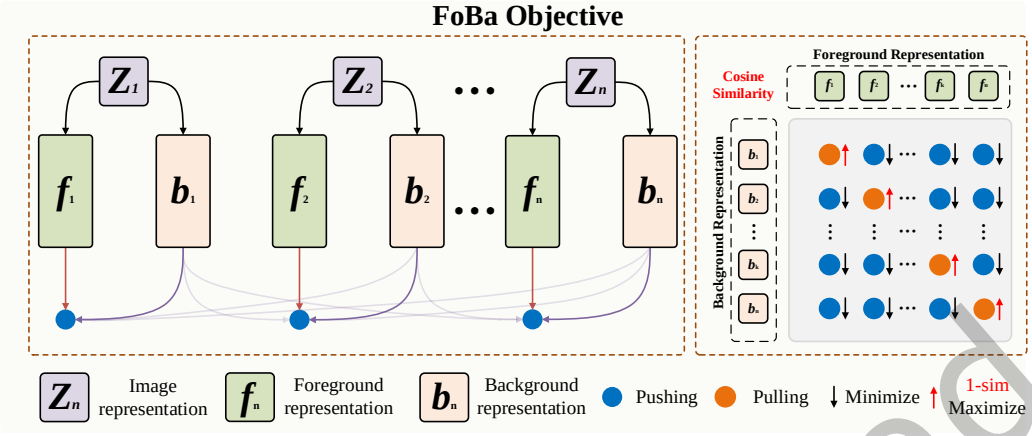


Fig. 8. Illustration of foreground-background contrastive learning. Given a batch with n images, each learned image representation Z_i is disentangled into foreground embedding f_i and background embedding b_i. We compute the cosine similarity between pairs, using 1 to subtract the matched foreground-background similarities as positive pairs and other unmatched similarities as negative pairs. In this manner, contrastive learning can push apart all foreground-background pairs.

Given n images from a batch, the foreground and background embeddings from the same image form a *positive pair*, and the foreground and background embeddings from different images form a *negative pair*. Then, we calculate the cosine similarity in positive and negative pairs:

$$\begin{cases} s_{i,i}^{P} = 1 - \text{sim}\left(\boldsymbol{f}_i, \boldsymbol{b}_i\right), \\ s_{i,j}^{N} = \text{sim}\left(\boldsymbol{f}_i, \boldsymbol{b}_j\right), \end{cases} \quad i, j \in \{1, \cdots, n\}, i \neq j \tag{12}$$

where $s_{i,i}^{P}$ and $s_{i,j}^{N}$ denote the cosine similarity of positive and negative pairs, respectively. $\text{sim}(\cdot)$ denotes the similarity function. Since contrastive learning maximizes the similarities of positive pairs while minimizing the similarities of negative pairs, it is unsuitable for our objective. Hence, we implement a simple operation that subtracts the matched foreground-background similarities from 1 to obtain the final positive pairs. Next, we propose our contrastive loss to achieve the FoBa objective:

$$\mathcal{L}_{FB} = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp\left(s_{i,i}^{P}/\tau\right)}{\exp\left(s_{i,i}^{P}/\tau\right) + \sum_{j=2}^{n} \exp\left(s_{i,j}^{N}/\tau\right)}, \tag{13}$$

where $\mathcal{L}_{FB}$ denotes the FoBa objective loss, and τ is a temperature hyper-parameter to control the smoothness of the exponential function. $\mathcal{L}_{FB}$ considers the foreground-background comparisons both intra-image $\left(s_{i,i}^{P}, i = i\right)$ and inter-images $\left(s_{i,j}^{N}, j \neq i\right)$ simultaneously.

3.7 Loss Function

Five predictions $\boldsymbol{P} = \{\boldsymbol{P}_i | i = 1, 2, 3, 4, 5\}$ and edge predictions $\boldsymbol{E} = \{\boldsymbol{E}_i | i = 1, 2, 3, 4, 5\}$ are our final outputs. Therefore, our loss function consists of three parts: the prediction loss $\mathcal{L}_P$, the edge loss $\mathcal{L}_E$, and the FoBa loss

$\mathcal{L}_{FB}$. The final loss $\mathcal{L}$ can be calculated as

$$\begin{cases} \mathcal{L}_P = \sum_{i=1}^{5} \mathcal{L}_{wbce}\left(\boldsymbol{P}_i, \boldsymbol{G}\right) + \mathcal{L}_{wiou}\left(\boldsymbol{P}_i, \boldsymbol{G}\right), \\ \mathcal{L}_E = \sum_{i=1}^{5} \mathcal{L}_{wbce}\left(\boldsymbol{E}_i, \boldsymbol{G}_E\right) + \mathcal{L}_{dice}\left(\boldsymbol{E}_i, \boldsymbol{G}_E\right), \\ \mathcal{L}_{FB} = \sum_{i=1}^{5} \mathcal{L}_{FB_i}, \\ \mathcal{L} = \mathcal{L}_P + \mathcal{L}_E + \mathcal{L}_{FB}, \end{cases} \tag{14}$$

where $\mathcal{L}_{wbce}(\cdot)$ and $\mathcal{L}_{wiou}(\cdot)$ denote the weighted binary CE loss and intersection-over-union (IoU) [64] loss, respectively. $\mathcal{L}_{dice}(\cdot)$ denotes the dice loss [43, 69] for edge supervision.

4 EXPERIMENTS

4.1 Settings

4.1.1 Datasets. We evaluate our proposed FBD-Net on four benchmark datasets: CHAMELEON [55], CAMO [31], COD10K [12], and NC4K [40]. Specifically, CHAMELEON is a small dataset that contains 76 camouflaged images. CAMO is the first proposed COD dataset, consisting of 1,250 camouflaged images and 1,250 non-camouflaged images. COD10K is the first complete large-scale camouflaged object dataset, consisting of 5,066 camouflaged images, 3,000 background images, and 1,934 non-camouflaged images. NC4K is the latest camouflaged object dataset, containing 4,121 images. Following the same settings [12], 1,000 images from CAMO and 3,040 images from COD10K are used for training, and the remaining images from each dataset are used for testing.

4.1.2 Evaluation metrics. Since COD is a class-agnostic task similar to the SOD task, the evaluation metrics of the Salient Object Detection (SOD) task are commonly used to evaluate COD models. Therefore, four evaluation metrics are used in the COD field, including S-measure (S_α) [10], weighted F-measure (F_β^ω) [41], mean absolute error (M) [49], and mean E-measure (E_ϕ) [11]. In addition, we also draw the precision-recall (**PR**) and F-measure curves to evaluate the performance of our model.

4.1.3 Implementation details. We implement PyTorch and two NVIDIA V100 GPUs with 32 GB memory on our model for training. We choose a CNN-based backbone (*i.e.,* Res2Net-50 [16]) and two representative Transformer-based backbones (*i.e.,* PVT-V2 [62] and Swin-V2 [39]) as our feature extractors, which have been pre-trained on ImageNet [7]. We use the Res2Net backbone as our experimental subject, and subsequent ablation experiments and hyper-parameters are explained based on Res2Net. We employ the AdamW [29] optimizer for training, and the batch size is set to 40. The initial learning rate is $8e-5$, and it follows the StepLR strategy, reducing to one-tenth of its original value every 50 epochs. Our model is trained for 150 epochs, and the input resolution is resized to 352×352. τ is set as 1 and the weight decay is 1e-4. The effect of input resolution (*e.g.,* 384×384, 416×416) will be provided in the subsequent experiments. Moreover, we adopt some data augmentation skills (*e.g.,* random flipping, random cropping) to enhance the generalization.

4.1.4 Comparison with state-of-the-art methods. To demonstrate the effectiveness of our model, we select 26 SOTA methods for comparisons, including EGNet [80], SINet [13], LSR [40], PFNet [42], UGTR [71], MGL [76], ERRNet [25], C^2FNet [5], SINetV2 [12], BSANet [86], FAPNet [84], Zoom-Net [46], SegMaR [26], PreyNet [78], FBNet [35], DGNet [24], GLNet [57], SARNet [70], MECS-Net [67], FSPNet [22], CamoFormer [73], DCNet [75], DRFNet [63], PRNet [21], PRBE-Net [74], and SENet [17]. Please note that these methods adopt different resolutions as inputs for their models and employ various backbones to extract features. Thus, the comparisons are relatively fair between them. Nonetheless, we also follow these existing works for comparison. The results of all methods are re-evaluated from the open prediction maps they provide (Using One-key evaluation MATLAB codes https://github.com/DengPingFan/CODToolbox).

4.2 Performance Comparison

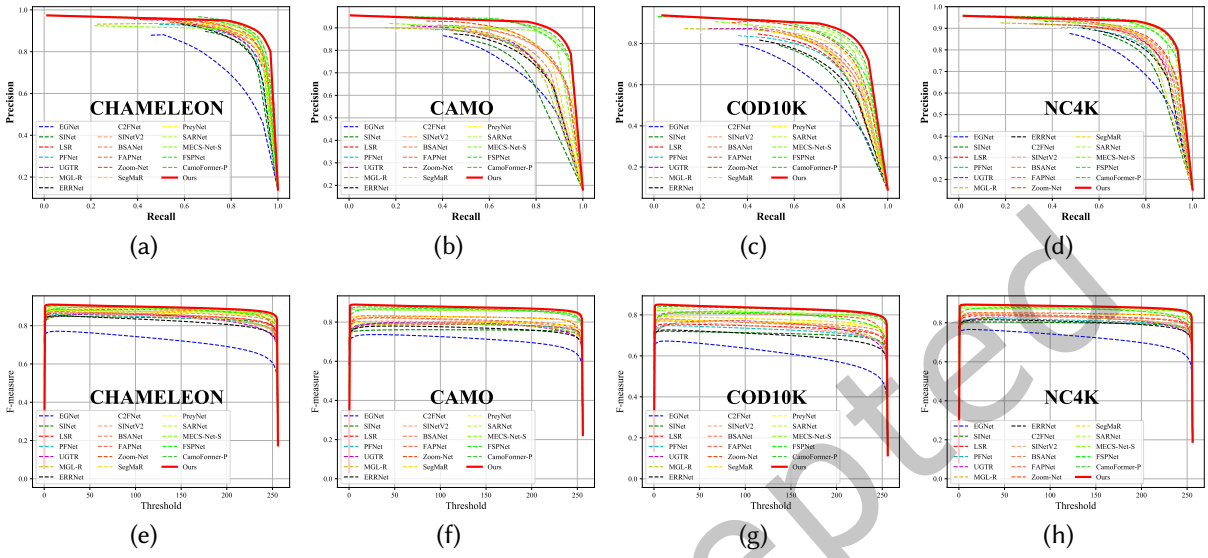


Fig. 9. PR (top) and F_measeure (bottom) curves of FBD-Net and 20 state-of-the-art models on the four datasets.

4.2.1 Quantitative comparison. Table 1 shows our quantitative results compared with 20 methods on four datasets under four common evaluation metrics. It can be seen that our FBD-Net outperforms all competitors, which demonstrates the effectiveness of our model. Specifically, Specifically, compared to FAPNet [84], which is based on the same Res2Net-50 backbone and input resolution 352×352, the Res2Net-50 version of FBD-Net achieves average improvements of 0.90%, 1.41%, 2.40%, and 7.33% in terms of S_α, E_ϕ, F_β^ω, and M, respectively. Notably, FAPNet employs a multi-scale training strategy to enhance generalizability, enabling it to learn high-resolution representations for improved performance. Moreover, compared with the Transformer-based SOTA methods (*i.e.,* SARNet [70] published in TCSVT2023, MECS-Net-S [67] published in SPL2023, FSPNet [22] published in CVPR2023), the PVT-V2-B4 version of FBD-Net shows significant improvements of 2.11%, 2.93%, 5.34%, and 10.92% over FSPNet [22] on average for S_α, E_ϕ, F_β^ω, and M, respectively. Notably, FSPNet has nearly four times the number of parameters compared to FBD-Net (274.240M *vs.* 68.699M). Furthermore, we observe that other Transformer-based versions of FBD-Net (*e.g.,* Swin-V2, PVT-V2-B2, B3) achieve excellent performance even at lower resolutions compared to other methods, highlighting that the performance gains are driven by the disentanglement and discrimination capabilities of our model.

Moreover, we also compare our method against the latest COD models such as SENet [17], PRBENet [74], PRNet [21], DCCFN [63], and LDRNet [75]. The results demonstrate that our method achieves consistently competitive performance across all benchmarks, and in many cases surpasses these newly published models. We would also like to clarify why we do not include certain other recent works (e.g., [79], [47], [38], and [56]) in our comparative experiments. These methods rely on significantly different learning paradigms, such as: Using Adapter structures [79], employing large input resolutions (e.g., 512×512) [79], utilizing multi-scale or multi-view inputs [47], [56]. As the COD task is known to be extremely sensitive to input resolution and feature resolution [70], these differences can lead to inherently unfair comparisons. In general, larger input sizes and multi-scale

Table 1. Comparisons with the recent 26 SOTAs for COD on four datasets in terms of four metrics. The best results are highlighted in **bold**. "-" denotes the results are not available. "**" means it adopts a **multi-scale training strategy or multi-view methods** to enhance the generalizability. "****" means it employs a **multi-stage detection fashion**. "↑" and "↓" mean that the results are better. "-R2": Res2Net-50. "-S2": Swin Transformer V2-B. "-P2": PVT-V2-B2. "-P3": PVT-V2-B3. "-P4": PVT-V2-B4.

Methods	Publication	Input size	Param (M)	FLOPs (G)	Backbone	CHAMELEON (76)				CAMO-Test (250)				COD10K-Test (2,026)				NC4K (4,121)			
						$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
CNN-Based Methods																					
EGNet [80]	ICCV2019	352 × 352	111.639	294.792	ResNet-50	0.797	0.860	0.649	0.065	0.732	0.800	0.604	0.109	0.736	0.810	0.517	0.061	0.777	0.841	0.639	0.075
SINet [13]	CVPR2020	352 × 352	48.947	19.553	ResNet-50	0.872	0.936	0.806	0.034	0.745	0.804	0.644	0.092	0.776	0.864	0.631	0.043	0.808	0.871	0.723	0.058
LSR [40]	CVPR2021	352 × 352	50.935	17.485	ResNet-50	0.890	0.935	0.822	0.030	0.787	0.838	0.696	0.080	0.804	0.880	0.673	0.037	0.840	0.895	0.766	0.048
PFNet [42]	CVPR2021	416 × 416	46.498	26.599	ResNet-50	0.882	0.931	0.810	0.033	0.782	0.842	0.695	0.085	0.800	0.877	0.660	0.040	0.829	0.888	0.745	0.053
UGTR** [71]	ICCV2021	473 × 473	48.868	121.658	ResNet-50	0.887	0.910	0.794	0.031	0.784	0.822	0.684	0.086	0.817	0.853	0.666	0.036	0.839	0.875	0.747	0.052
MGL-R** [76]	CVPR2021	473 × 473	67.636	**431.869**	ResNet-50	0.893	0.918	0.813	0.030	0.775	0.812	0.673	0.088	0.814	0.852	0.666	0.035	0.833	0.867	0.740	0.052
ERRNet** [25]	PR2022	352 × 352	67.757	20.090	ResNet-50	0.868	0.922	0.787	0.039	0.779	0.842	0.679	0.085	0.786	0.867	0.630	0.043	0.827	0.887	0.737	0.054
C2FNet** [5]	TCSVT2022	352 × 352	25.214	18.190	Res2Net-50	0.893	0.946	0.845	0.028	0.799	0.859	0.730	0.077	0.811	0.887	0.691	0.036	0.840	0.896	0.770	0.048
SINetV2[12]	TPAMI2022	352 × 352	26.976	12.313	Res2Net-50	0.888	0.942	0.816	0.030	0.820	0.882	0.743	0.070	0.815	0.887	0.680	0.037	0.847	0.903	0.770	0.048
BSANet [86]	AAAI2022	384 × 384	32.585	29.764	Res2Net-50	0.895	0.946	0.841	0.027	0.794	0.851	0.717	0.079	0.818	0.891	0.699	0.034	0.841	0.897	0.771	0.048
FAPNet** [84]	TIP2022	352 × 352	29.524	29.732	Res2Net-50	0.893	0.940	0.825	0.028	0.815	0.865	0.734	0.076	0.822	0.888	0.694	0.036	0.851	0.899	0.775	0.047
Zoom-Net**[46]	CVPR2022	384 × 384	32.400	101.800	ResNet-50	0.902	0.943	0.845	0.023	0.820	0.878	0.752	0.066	0.838	0.888	0.729	0.029	0.853	0.896	0.784	0.043
SegMaR**** [26]	CVPR2022	352 × 352	56.215	33.654	ResNet-50	0.906	0.951	0.860	0.025	0.815	0.874	0.753	0.071	0.833	0.899	0.724	0.034	0.841	0.896	0.781	0.046
PreyNet[78]	MM2022	448 × 448	38.534	58.256	ResNet-50	0.895	0.952	0.844	0.028	0.790	0.842	0.708	0.077	0.813	0.881	0.697	0.034	-	-	-	-
FBNet[35]	TOMM2022	352 × 352	-	-	ResNet-50	0.888	0.939	0.828	0.032	0.783	0.839	0.702	0.081	0.809	0.889	0.684	0.035	-	-	-	-
DGNet[24]	MIR2023	352 × 352	-	-	EfficientNet	-	-	-	-	0.839	0.901	0.769	0.057	0.822	0.896	0.693	0.033	0.857	0.911	0.784	0.042
GLNet[57]	TITS2023	**704 × 704**	-	-	EfficientNet	0.919	0.963	0.881	0.021	0.851	0.901	0.799	0.059	0.871	0.930	0.794	0.023	0.876	0.921	0.827	0.037
Ours-R2		352 × 352	32.735	29.408	Res2Net-50	0.896	0.940	0.836	0.027	0.829	0.886	0.761	0.066	0.831	0.898	0.715	0.033	0.855	0.905	0.787	0.045
Ours-R2		384 × 384	32.735	-	Res2Net-50	0.910	0.951	0.858	0.027	0.830	0.882	0.761	0.068	0.837	0.900	0.724	0.032	0.860	0.909	0.795	0.043
Ours-R2		416 × 416	32.735	-	Res2Net-50	0.903	0.944	0.848	0.027	0.820	0.872	0.751	0.075	0.840	0.899	0.728	0.032	0.862	0.909	0.797	0.042
Ours-R2		448 × 448	32.735	-	Res2Net-50	0.902	0.939	0.848	0.028	0.827	0.878	0.761	0.071	0.843	0.903	0.735	0.031	0.861	0.908	0.797	0.043
Ours-R2		480 × 480	32.735	-	Res2Net-50	0.902	0.944	0.851	0.028	0.818	0.867	0.750	0.075	0.845	0.904	0.741	0.031	0.861	0.906	0.796	0.044
Transformer-Based Methods																					
SARNet[70]	TCSVT2023	384 × 384	47.200	23.100	PVT-V2-B3	0.912	0.957	0.871	0.021	0.868	0.927	0.828	0.047	0.864	0.931	0.777	0.024	0.886	0.937	0.842	0.032
MECS-Net-S[67]	SPL2023	384 × 384	119.258	135.273	Swin-V2-B	0.901	0.953	0.857	0.023	0.853	0.911	0.811	0.051	0.854	0.919	0.770	0.026	0.874	0.924	0.829	0.036
FSPNet[22]	CVPR2023	384 × 384	**274.240**	283.311	ViT-B	0.908	0.943	0.851	0.023	0.856	0.899	0.799	0.050	0.851	0.895	0.735	0.026	0.879	0.915	0.816	0.035
CamoFormer[73]	TPAMI2024	384 × 384	71.403	47.269	PVT-V2-B4	0.910	0.957	0.865	0.022	0.872	0.929	0.831	0.046	0.869	0.932	0.786	0.023	0.892	0.939	0.847	0.030
DCNet [75]	TCSVT2023	480 × 480	54.43	94.74	PVT-V2-B4	0.920	0.958	0.890	0.019	0.870	0.922	0.831	0.050	0.873	0.934	0.810	0.022	-	-	-	-
DRFNet [63]	TCSVT2024	416 × 416	-	-	MiT	0.918	0.959	0.880	0.019	0.868	0.925	0.832	0.047	0.869	0.936	0.792	0.023	0.887	0.939	0.846	0.031
PRNet [21]	TCSVT2024	384 × 384	-	-	PVT-V2-B4	0.909	0.957	0.876	0.024	0.860	0.915	**0.848**	0.050	0.859	0.925	0.793	0.025	0.879	0.929	0.858	0.035
PRBE-Net[74]	TMM2025	384 × 384	-	-	PVT-V2-B4	0.918	0.951	0.878	0.020	0.876	0.928	0.837	0.045	0.867	0.932	0.793	0.021	0.887	0.931	0.845	0.031
SENet[17]	TIP2025	384 × 384	-	88.5	ViT	**0.918**	0.957	**0.878**	**0.019**	**0.888**	0.932	0.847	**0.039**	0.865	0.925	0.780	0.024	0.889	0.933	0.843	0.032
Ours-P3		352 × 352	51.382	34.377	PVT-V2-B3	0.913	0.956	0.866	0.022	0.875	0.928	0.832	0.044	0.870	0.930	0.784	0.023	0.888	0.935	0.842	0.032
Ours-P4		352 × 352	68.699	42.076	PVT-V2-B4	0.913	**0.961**	0.869	0.023	0.879	**0.933**	0.841	0.042	0.870	0.931	0.786	0.023	0.891	0.938	0.849	0.031
Ours-S2		384 × 384	94.690	55.914	Swin-V2-B	0.910	0.949	0.862	0.023	0.882	0.932	0.842	0.041	0.873	0.930	0.791	0.023	0.890	0.935	0.846	0.032
Ours-P4		384 × 384	68.699	50.074	PVT-V2-B4	0.916	0.953	0.875	0.023	0.881	0.931	0.844	0.043	**0.877**	**0.935**	**0.799**	**0.022**	**0.893**	**0.939**	**0.852**	**0.030**

Table 2. Ablation studies for these modules and components.

Methods	Components	Param (M)	FLOPs (G)	CAMO-Test (250)				COD10K-Test (2,026)			
				$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$
①	w/o CAM	30.752	24.495	0.824	0.878	0.756	0.069	0.828	0.898	0.712	0.033
②	w/o SIEP	31.674	25.750	0.819	0.873	0.749	0.069	0.827	0.894	0.707	0.033
③	w/o EFBD	29.650	21.461	0.818	0.870	0.749	0.073	0.824	0.890	0.702	0.035
④	w/o CSAF	31.702	26.754	0.819	0.871	0.748	0.070	0.826	0.894	0.706	0.034
⑤	w/o DisHead	31.987	29.075	0.821	0.876	0.750	0.069	0.829	0.897	0.712	0.033
Ours		32.735	29.408	**0.829**	**0.886**	**0.761**	**0.066**	**0.831**	**0.898**	**0.715**	**0.033**

inputs tend to yield better results, which could give those methods an advantage that is unrelated to their core model design. Despite using a relatively simpler backbone, our method still outperforms the latest ViT-based competitors [17] on large-scale datasets such as COD10K and NC4K, which further validates the effectiveness and efficiency of our approach.

In addition to these quantitative comparisons regarding the four metrics, we also show PR and F-measure curves in Fig. 9. The highest curve indicates that the corresponding model achieves the best performance. Thus, it can be seen that our model achieves the best results.

4.2.2 Complexity comparison. We list the number of parameters (Params) and the floating point operations (FLOPs) of each model in the fourth and fifth columns of Table 1. The results show that both the Res2Net-50 and Transformer versions of FBD-Net achieve better performance with relatively smaller parameters than these representative COD models. Meanwhile, it can be seen that although the SOTA model FSPNet achieves superior performance, the model complexity it consumes is also expensive. Also, our method maintains a relatively moderate complexity level. For instance, our Transformer-based variant has only 68.699M parameters and 50.074 GFLOPs, which is significantly lower than SENet [17] (published in TIP2025), which requires 88.5 GFLOPs. This comparison highlights the efficiency of our design while still achieving strong performance.

4.2.3 Qualitative comparison. Fig. 10 presents the visual results in comparison with 20 state-of-the-art methods across several challenging scenarios. From these comparisons, it is evident that our model demonstrates superior robustness. To be specific, in the cases of small objects (1^{st}-2^{nd} rows) and large objects (3^{rd}-4^{th} rows), our model effectively detects objects across varying scales. Moreover, in scenarios involving multiple objects (5^{th}-6^{th} rows) and occlusion objects (7^{th}-8^{th} rows), our model excels in accurate object localization, maintaining structural integrity, and producing clearer prediction maps. Regarding the more challenging scenario of uncertainty edge objects (9^{th}-10^{th} rows), our model demonstrates significant advantages.

4.3 Ablation Study

In this subsection, we conduct a series of ablation studies investigating the effectiveness of our proposed modules and components. Note that all ablation studies are based on the Res2Net-50 backbone for convenience. The results of all ablation studies are shown in Table 2.

4.3.1 Effectiveness of CAM module. To verify the effectiveness of the CAM, we remove it from the FBD-Net and keep the rest of the parts intact. The ablation results of the module are shown in the 2^{rd} row of Table 2 (denoted

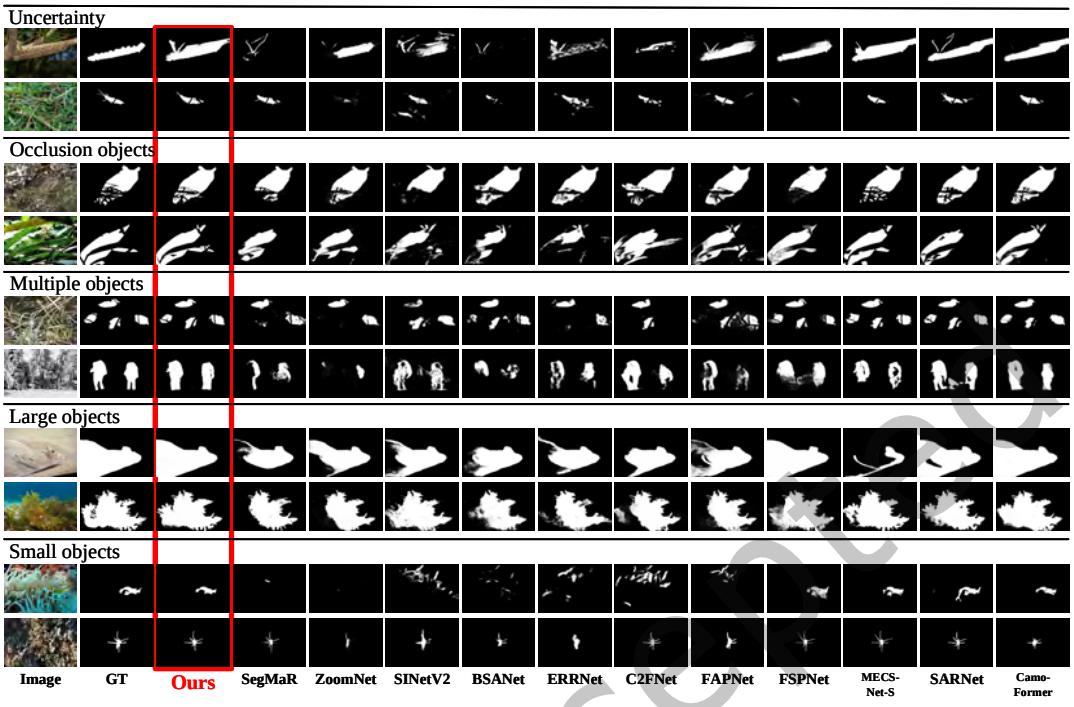


Fig. 10. Visual comparison of some representative COD models and ours in different scenarios.

as ①). Compared with 'Ours,' the performance ① is slightly degraded (*e.g.*, S_α: 0.831 → 0.828 on COD10K, S_α: 0.829 → 0.824 on CAMO). The results demonstrate that our CAM can aggregate context coarse-grained semantics for COD performance improvement.

4.3.2 Effectiveness of SIEP module. To illustrate the effectiveness of the SIEP module, we ablate it from the entire FBD-Net and show the results in the 3^{rd} row of Table 2 (denoted as ②). It can be observed that SIEP can improve performance, which demonstrates that introducing multi-scale features and inter-scale interactions are effective for COD (*e.g.*, F_β^ω: 0.707 → 0.715 on COD10K, F_β^ω: 0.749 → 0.761 on CAMO). The reason behind performance improvement is that SIEP accumulates subtle but effective multi-scale camouflaged cues for COD.

4.3.3 Effectiveness of EFBD module. To evaluate the effectiveness of the proposed EFBD module, we conducted ablation studies by removing it from the complete FBD-Net. The results, presented in the 4^{th} row of Table 2 (denoted as ③), clearly demonstrate that EFBD significantly enhances performance (*e.g.*, F_β^ω: 0.702 → 0.715 on COD10K, F_β^ω: 0.749 → 0.761 on CAMO). These improvements underscore the critical role EFBD plays in our model. However, EFBD introduces additional parameters and FLOPs (*e.g.*, Params: 29.650 → 32.735, FLOPs: 21.461 → 29.408), highlighting the need for a more lightweight EFBD module as a future development goal. In summary, the results further validate the effectiveness of our approach, emphasizing the importance of a specially designed edge-guided decoupling module in aiding the detection of imperceptible camouflaged objects. The separated learning mechanism enhances the understanding of camouflage features.

In addition, in Fig. 11, we leverage the Grad-CAM [53] algorithm to visualize the output attention maps of the EFBD module. These visualizations reveal that the EFBD module progressively focuses on finer edge details

as decoding deepens. The results demonstrate that the EFBD module effectively captures more discriminative semantic features, highlighting its contribution to the overall performance of our model.

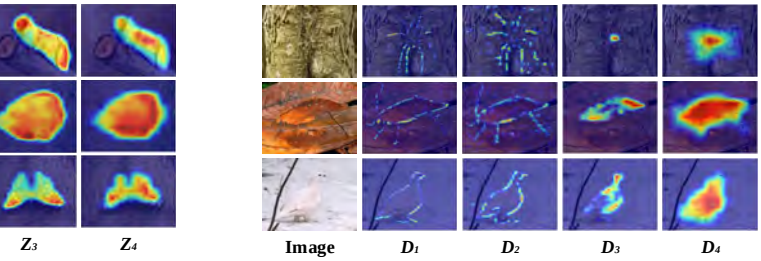


Fig. 11. the focus areas of our EFBD modules (from D_4 to D_1).

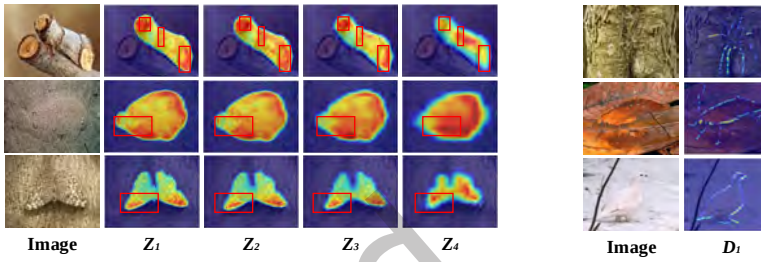


Fig. 12. the focus areas of our CSAF modules (from Z_4 to Z_1).

4.3.4 Effectiveness of CSAF module. To evaluate the effectiveness of the proposed CSAF module, we conducted ablation studies by removing it from the complete model and replacing it with additional operations for fusing cross-stage features. The results, shown in the 5^{th} row of Table 2 (denoted as ④), highlight the advantages of our CSAF module. Specifically, comparisons between ④ and Ours (*e.g.*, M: 0.034 → 0.033 on COD10K, M: 0.070 → 0.066 on CAMO), demonstrate that the CSAF module outperforms simple addition operations, confirming the importance of a selective cross-stage feature fusion strategy for COD. Furthermore, the CSAF module effectively adapts to select and accumulate camouflaged cues, contributing to performance enhancement. To illustrate the role of the CSAF module more intuitively, we use the Grad-CAM [53] algorithm to visualize its output attention maps in Fig. 12. The visualizations indicate that the CSAF module focuses on more subtle cues, further validating its utility.

4.3.5 Effectiveness of DisHead. To prove the role of our proposed DisHead, we also conduct ablation studies by removing it from the entire FBD-Net and show the results in the 6^{th} row of Table 2 (denoted as ⑤). From the quantitative comparison results (*e.g.*, F_β^ω: 0.712 → 0.715 on COD10K, F_β^ω: 0.750 → 0.761 on CAMO), which shows that DisHead can improve detection performance. We attribute this improvement to the fact that DisHead can boost the disentangling ability of foreground and background representations, further improving detection performance. Here, we show several learned edge prediction maps in Fig. 13. From these edge cases, we can see that our model can learn better edge representations for the DisHead to enhance discriminative representations.

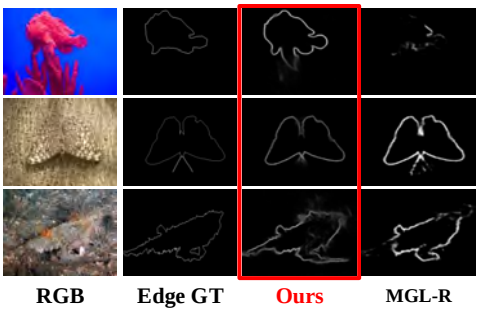
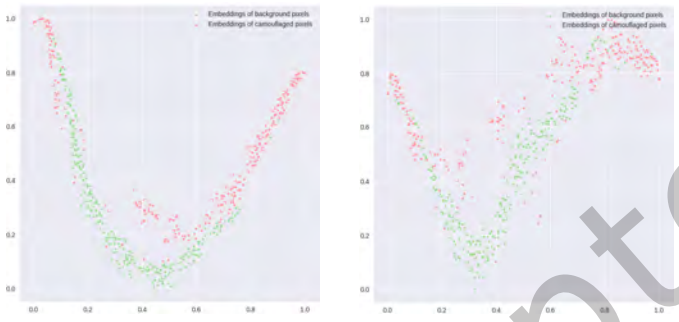


Fig. 13. Edge prediction maps of camouflaged objects.

Additionally, to provide a more intuitive understanding of the role of DisHead, we utilize the t-SNE algorithm [61] to visualize the learned feature distributions. Following the method outlined in [66], we calculate the average embeddings of foreground and background pixels for each sample. As illustrated in Fig. 14, the distributions of camouflage and background embeddings show that embeddings with the same label are more tightly clustered when DisHead is employed. This observation highlights that our proposed DisHead effectively enhances the discriminative power of feature representations.



(a) with DisHead (b) without DisHead

Fig. 14. T-SNE visualization of learned camouflaged representations of CAMO dataset. (a) with DisHead. (b) without DisHead. Embeddings are colored according to labels.

4.3.6 Impact of Input Resolution on Performance. COD is generally sensitive to input resolution, with larger resolutions typically yielding better performance [20, 70]. Most researchers optimize their models by training with varying resolutions to achieve the best results. To investigate this, we evaluate the impact of multiple input resolutions (*i.e.,* 384 × 384, 416 × 416, 448 × 448, and 480 × 480) on performance, with experimental results presented in Table 1. Due to GPU memory constraints, different batch sizes are used for each resolution. Specifically, comparing these results with our baseline resolution (352 × 352), we observe that higher resolutions generally improve performance. Larger input resolutions enable the model to capture finer details, enhancing its effectiveness (*e.g.,* GLNet [57] produces superior performance using a large input resolution of **704** × **704**.). Consequently, we recommend that future researchers adopt higher input resolutions for COD tasks, provided adequate GPU resources are available.

4.4 Application to polyp segmentation

Polyp segmentation, as one of the downstream tasks of COD, often has a similar pattern to camouflaged objects. Therefore, taking the proposed model to adaptation polyp segmentation is an important measurement. To this end, we use four polyp segmentation datasets, including CVC-ClinicDB [2], CVC-ColonDB [60], ETIS [54], and Kvasir [23], to evaluate our model. We follow the existing protocol [14, 15, 28, 48] to train our model, in which 900 images from Kvasir and 550 images from CVC-ClinicDB are used for training. The remaining images are used for testing. We use PVT-V2 as our backbone, and the input resolution is 352 × 352. Five SOTA methods are used for comparison, namely, U-Net [52], U-Net++ [85], SFA [15], PraNet [14], and UACANet-S [28]. In addition, we apply six metrics to evaluate results, including mDice, mIoU, S-measure (S_α) [10], weighted F-measure (F_β^ω) [41], mean absolute error (M) [49], and mean E-measure (E_ϕ). [11].

Table 3. Comparisons with the recent five SOTAs for polyp segmentation on four datasets in terms of six metrics. The best results are highlighted in **bold**. "↑" and "↓" mean that the results are better.

CVC-ClinicDB

Methods	mDice↑	mIoU↑	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
U-Net	0.823	0.755	0.889	0.913	0.811	0.019
U-Net++	0.794	0.729	0.873	0.891	0.785	0.022
SFA	0.700	0.607	0.793	0.840	0.647	0.042
PraNet	0.899	0.849	0.936	0.963	0.896	0.009
UACANet-S	0.916	0.870	0.939	0.965	**0.917**	0.008
Ours	**0.917**	**0.870**	**0.945**	**0.966**	0.910	**0.008**

CVC-ColonDB

Methods	mDice↑	mIoU↑	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
U-Net	0.512	0.444	0.712	0.696	0.498	0.061
U-Net++	0.483	0.410	0.691	0.680	0.467	0.064
SFA	0.469	0.347	0.634	0.675	0.379	0.094
PraNet	0.712	0.640	0.820	0.847	0.699	0.043
UACANet-S	0.783	0.704	0.847	**0.894**	**0.772**	**0.034**
Ours	**0.784**	**0.713**	**0.860**	0.876	0.764	0.036

ETIS

Methods	mDice↑	mIoU↑	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
U-Net	0.398	0.335	0.684	0.643	0.366	0.036
U-Net++	0.401	0.344	0.683	0.629	0.390	0.035
SFA	0.297	0.217	0.557	0.531	0.231	0.109
PraNet	0.628	0.567	0.794	0.808	0.600	0.031
UACANet-S	0.694	0.615	0.815	0.848	0.650	0.023
Ours	**0.794**	**0.718**	**0.881**	**0.905**	**0.755**	**0.017**

Kvasir

Methods	mDice↑	mIoU↑	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$
U-Net	0.818	0.746	0.858	0.881	0.794	0.055
U-Net++	0.821	0.743	0.862	0.886	0.808	0.048
SFA	0.723	0.611	0.782	0.834	0.670	0.075
PraNet	0.898	0.840	0.915	0.944	0.885	0.030
UACANet-S	0.905	0.852	0.914	0.948	0.897	0.026
Ours	**0.913**	**0.862**	**0.927**	**0.963**	**0.910**	**0.021**

4.4.1 Quantitative comparison. Table 3 shows the quantitative results of these polyp segmentation methods. It can be clearly seen that our model achieves the best performance compared with these methods, which demonstrates that our model can be effectively adapted to polyp segmentation. Especially in terms of ETIS and Kvasir datasets, the five metrics have significantly improved (*e.g.,* mDice: 0.694 → 0.794 on ETIS, mDice: 0.905 → 0.913 on Kvasir). Hence, we believe that developing a general model to bridge the gap between COD and polyp segmentation is an interesting and worthwhile direction in the future.

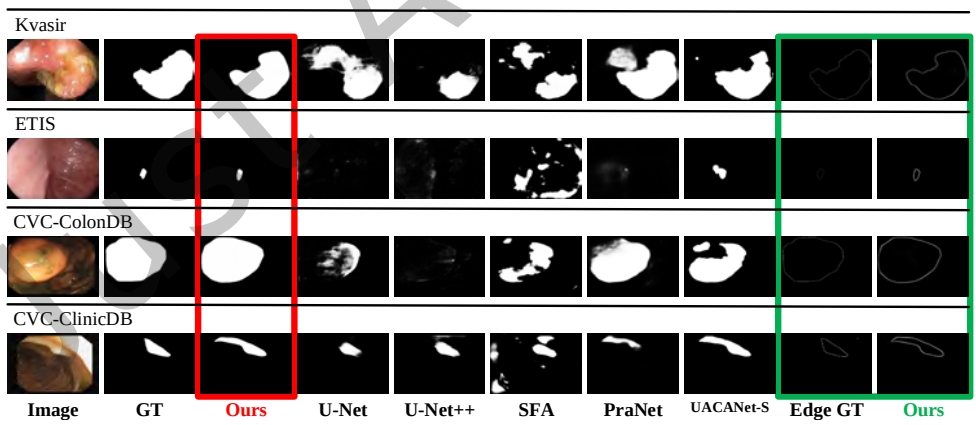


Fig. 15. Visual comparison of some representative polyp segmentation models and ours in different scenarios.

4.4.2 Qualitative comparison. Fig. 15 shows the visual comparison results. Specifically, we select one case from each dataset to show the generalization of our model. Our model performs better in small object cases (1^{st} and 3^{rd}

rows) and large objects (2^{nd} and 4^{th} rows) compared with other methods. Furthermore, our model can precisely learn polyp edge representations (last column). Therefore, our model can be successfully employed in polyp segmentation tasks.

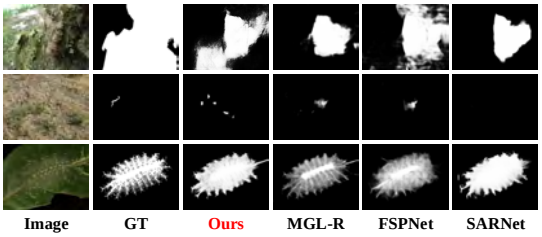


Fig. 16. Illustration of failure cases.

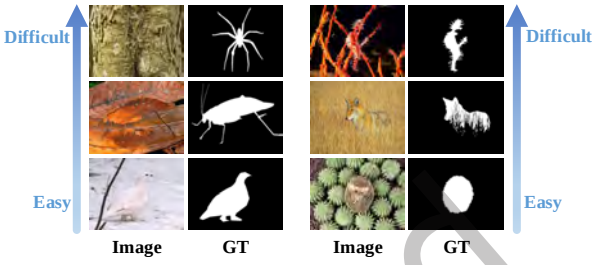


Fig. 17. Recognition levels of different camouflaged objects, from easy to difficult.

4.5 Discussion

4.5.1 Failure cases. Although our proposed FBD-Net achieves superior performance, it also outputs wrong prediction maps in some cases. Here, we show several failure cases in Fig. 16 above. Specifically, when meeting objects that uncertainly edge and contain intricate internal details, FBD-Net may fail to detect the entire object and edge (see 1^{st} row). Moreover, when meeting tiny objects (see 2^{nd} row) and high similarity with surrounding objects (see 3^{rd} row), our method still has difficulty identifying objects. Similarly, some representative SOTA methods, such as the edge-guided method MGL-R [76] and other methods [22, 70], also fail to tackle the scenarios. We attribute these failures to the limited ability of these methods to capture fine-grained discriminative features.

4.5.2 Limitations and future Works. We have identified four shortcomings that deserve further exploration in the future. Firstly, we aim to enhance COD by decoupling foreground and background representations. However, our proposed method is limited in the semantic-level aspect. For example, our proposed FoBa Objective only considers semantic-level contrast but ignores fine-grained information contrast, which may generate the wrong results in Fig. 16. In a word, we will explore a multi-view FoBa Objective to achieve better performance. Secondly, we ignore camouflaged levels of different objects. For instance, in Fig. 17, intuitively, the white bird in the first row is more conspicuous and easier to identify than the spider in the third row. This is due to the higher camouflaged level in the third row. Therefore, we will design an adaptive module that can enhance the understanding of objects with different camouflage levels. Thirdly, while we are the first to attempt to enhance camouflage discrimination for COD, our research in this area is still in its early stages due to the abstract nature of the concept. Looking ahead, we plan to incorporate textual knowledge to enhance the understanding of camouflage knowledge, for instance, by leveraging large models like CLIP [51]. Finally, there is no research on camouflaged objects in traffic scenes. This limitation is mainly due to the lack of large-scale datasets for traffic camouflage scenes. In the future, we plan to develop a comprehensive dataset of traffic camouflage scenes to promote further research in this field.

5 CONCLUSION

In this paper, we propose FBD-Net, a novel approach to enhance Camouflaged Object Detection (COD). Unlike existing methods, our approach focuses on enhancing foreground-background disentanglement to boost detection capabilities. We introduce the EFBD module to facilitate decoupling and separate learning for COD, alongside the DisHead, which strengthens the discriminative power of our model. Additionally, we incorporate the CAM,

SIEP, and CSAF modules to achieve initial object detection, multi-scale information extraction, and subtle clue accumulation, respectively. Looking ahead, we plan to develop a large-scale dataset specifically tailored to traffic camouflage scenarios to support future research.

REFERENCES

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35, 8 (2013), 1798–1828.

[2] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. 2015. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized medical imaging and graphics* 43 (2015), 99–111.

[3] Nagappa U Bhajantri and P Nagabhushan. 2006. Camouflage defect identification: a novel approach. In *9th International Conference on Information Technology (ICIT'06)*. IEEE, 145–148.

[4] Yuxuan Cai, Yizhuang Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Li, and Xiangyu Zhang. 2022. Reversible column networks. *arXiv preprint arXiv:2212.11696* (2022).

[5] Geng Chen, Si-Jie Liu, Yu-Jia Sun, Ge-Peng Ji, Ya-Feng Wu, and Tao Zhou. 2022. Camouflaged object detection via context-aware cross-level fusion. *IEEE Transactions on Circuits and Systems for Video Technology* 32, 10 (2022), 6981–6993.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[8] Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. 2012. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474* (2012).

[9] Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. 2018. Large margin deep networks for classification. *Advances in neural information processing systems* 31 (2018).

[10] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. 2017. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*. 4548–4557.

[11] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. 2018. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421* (2018).

[12] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. 2021. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 10 (2021), 6024–6042.

[13] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. 2020. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2777–2787.

[14] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. 2020. Pranet: Parallel reverse attention network for polyp segmentation. In *International conference on medical image computing and computer-assisted intervention*. Springer, 263–273.

[15] Yuqi Fang, Cheng Chen, Yixuan Yuan, and Kai-yu Tong. 2019. Selective feature aggregation network with area-boundary constraints for polyp segmentation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part I 22*. Springer, 302–310.

[16] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. 2019. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence* 43, 2 (2019), 652–662.

[17] Chao Hao, Zitong Yu, Xin Liu, Jun Xu, Huanjing Yue, and Jingyu Yang. 2025. A Simple Yet Effective Network Based on Vision Transformer for Camouflaged Object and Salient Object Detection. *IEEE Transactions on Image Processing* 34 (2025), 608–622.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[19] Geoffrey Hinton. 2023. How to represent part-whole hierarchies in a neural network. *Neural Computation* 35, 3 (2023), 413–452.

[20] Xiaobin Hu, Shuo Wang, Xuebin Qin, Hang Dai, Wenqi Ren, Donghao Luo, Ying Tai, and Ling Shao. 2023. High-resolution iterative feedback network for camouflaged object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 881–889.

[21] Xihang Hu, Xiaoli Zhang, Fasheng Wang, Jing Sun, and Fuming Sun. 2024. Efficient Camouflaged Object Detection Network Based on Global Localization Perception and Local Guidance Refinement. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 7 (2024), 5452–5465.

[22] Zhou Huang, Hang Dai, Tian-Zhu Xiang, Shuo Wang, Huai-Xin Chen, Jie Qin, and Huan Xiong. 2023. Feature shrinkage pyramid for camouflaged object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5557–5566.

[23] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. 2020. Kvasir-seg: A segmented polyp dataset. In *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26*. Springer, 451–462.
[24] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. 2023. Deep gradient learning for efficient camouflaged object detection. *Machine Intelligence Research* 20, 1 (2023), 92–108.
[25] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu. 2022. Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognition* 123 (2022), 108414.
[26] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. 2022. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4713–4722.
[27] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems* 33 (2020), 18661–18673.
[28] Taehun Kim, Hyemin Lee, and Daijin Kim. 2021. Uacanet: Uncertainty augmented context attention for polyp segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2167–2175.
[29] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[30] Hala Lamdouar, Weidi Xie, and Andrew Zisserman. 2023. The making and breaking of camouflage. In *Proceedings of the IEEE/CVF international conference on computer vision*. 832–842.
[31] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. 2019. Anabranch network for camouflaged object segmentation. *Computer vision and image understanding* 184 (2019), 45–56.
[32] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. 2021. Uncertainty-aware joint salient object and camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10071–10081.
[33] Peng Li, Xuefeng Yan, Hongwei Zhu, Mingqiang Wei, Xiao-Ping Zhang, and Jing Qin. 2022. FindNet: Can You Find Me? Boundary-and-Texture Enhancement Network for Camouflaged Object Detection. *IEEE Transactions on Image Processing* 31 (2022), 6396–6411.
[34] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. 2019. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 510–519.
[35] Jiaying Lin, Xin Tan, Ke Xu, Lizhuang Ma, and Rynson WH Lau. 2023. Frequency-aware camouflaged object detection. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 2 (2023), 1–16.
[36] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. 2016. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295* (2016).
[37] Yu Liu, Haihang Li, Juan Cheng, and Xun Chen. 2023. MSCAF-Net: A General Framework for Camouflaged Object Detection via Learning Multi-Scale Context-Aware Features. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 9 (2023), 4934–4947.
[38] Zijian Liu, Xiaoheng Deng, Ping Jiang, Conghao Lv, Geyong Min, and Xin Wang. 2024. Edge Perception Camouflaged Object Detection Under Frequency Domain Reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 10 (2024), 10194–10207.
[39] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. 2022. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12009–12019.
[40] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. 2021. Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11591–11601.
[41] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. 2014. How to evaluate foreground maps?. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 248–255.
[42] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. 2021. Camouflaged object segmentation with distraction mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8772–8781.
[43] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*. Ieee, 565–571.
[44] Yuzhen Niu, Lifen Yang, Rui Xu, Yuezhou Li, and Yuzhong Chen. 2024. Minet: Weakly-supervised camouflaged object detection through mutual interaction between region and edge cues. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 6316–6325.
[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).
[46] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. 2022. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2160–2170.
[47] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. 2024. ZoomNeXt: A Unified Collaborative Pyramid Network for Camouflaged Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 9205–9220.
[48] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu. 2020. Multi-scale interactive network for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9413–9422.
[49] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. 2012. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 733–740.

[50] Ricardo Pérez-de la Fuente, Xavier Delclòs, Enrique Peñalver, Mariela Speranza, Jacek Wierzchos, Carmen Ascaso, and Michael S Engel. 2012. Early evolution and ecology of camouflage in insects. *Proceedings of the National Academy of Sciences* 109, 52 (2012), 21414–21419.
[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.
[53] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.
[54] Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. 2014. Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery* 9 (2014), 283–293.
[55] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. 2018. Animal camouflage analysis: Chameleon database. *Unpublished manuscript* 2, 6 (2018), 7.
[56] Xiaogang Song, Pengfei Zhang, Xiaofeng Lu, Xinhong Hei, and Rongrong Liu. 2024. A Universal Multi-View Guided Network for Salient Object and Camouflaged Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 11 (2024), 11184–11197.
[57] Bangyong Sun, Ming Ma, Nianzeng Yuan, Junhuai Li, and Tao Yu. 2024. Detecting the Background-Similar Objects in Complex Transportation Scenes. *IEEE Transactions on Intelligent Transportation Systems* 25, 3 (2024), 2920–2932.
[58] Shizhao Sun, Wei Chen, Liwei Wang, and Tie-Yan Liu. 2015. Large margin deep neural networks: Theory and algorithms. *arXiv preprint arXiv:1506.05232* 148 (2015).
[59] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. 2022. Boundary-guided camouflaged object detection. *arXiv preprint arXiv:2207.00794* (2022).
[60] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. 2015. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging* 35, 2 (2015), 630–644.
[61] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
[62] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media* 8, 3 (2022), 415–424.
[63] Yongchao Wang, Xiuli Bi, Bo Liu, Yang Wei, Weisheng Li, and Bin Xiao. 2024. Learning Discriminative Representations From Cross-Scale Features for Camouflaged Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 12 (2024), 12756–12769.
[64] Jun Wei, Shuhui Wang, and Qingming Huang. 2020. F^3Net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 12321–12328.
[65] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.
[66] Jiesheng Wu, Fangwei Hao, Weiyun Liang, and Jing Xu. 2024. Transformer Fusion and Pixel-Level Contrastive Learning for RGB-D Salient Object Detection. *IEEE Transactions on Multimedia* 26 (2024), 1011–1026.
[67] Jiesheng Wu, Weiyun Liang, Fangwei Hao, and Jing Xu. 2023. Mask-and-Edge Co-Guided Separable Network for Camouflaged Object Detection. *IEEE Signal Processing Letters* 30 (2023), 748–752.
[68] Zhe Wu, Li Su, and Qingming Huang. 2019. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3907–3916.
[69] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. 2020. Segmenting transparent objects in the wild. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer, 696–711.
[70] Haozhe Xing, Shuyong Gao, Yan Wang, Xujun Wei, Hao Tang, and Wenqiang Zhang. 2023. Go Closer to See Better: Camouflaged Object Detection via Object Area Amplification and Figure-Ground Conversion. *IEEE Transactions on Circuits and Systems for Video Technology* 33, 10 (2023), 5444–5457.
[71] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. 2021. Uncertainty-guided transformer reasoning for camouflaged object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4146–4155.
[72] Yang Yang and Qiang Zhang. 2024. Finding Camouflaged Objects Along the Camouflage Mechanisms. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 4 (2024), 2346–2360.
[73] Bowen Yin, Xuying Zhang, Deng-Ping Fan, Shaohui Jiao, Ming-Ming Cheng, Luc Van Gool, and Qibin Hou. 2024. CamoFormer: Masked Separable Attention for Camouflaged Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 10362–10374.
[74] Guanghui Yue, Shangjie Wu, Tianwei Zhou, Gang Li, Jie Du, Yu Luo, and Qiuping Jiang. 2025. Progressive Region-to-Boundary Exploration Network for Camouflaged Object Detection. *IEEE Transactions on Multimedia* 27 (2025), 236–248.

[75] Guanghui Yue, Houlu Xiao, Hai Xie, Tianwei Zhou, Wei Zhou, Weiqing Yan, Baoquan Zhao, Tianfu Wang, and Qiuping Jiang. 2023. Dual-constraint coarse-to-fine network for camouflaged object detection. *IEEE Transactions on Circuits and Systems for Video Technology* 34, 5 (2023), 3286–3298.
[76] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. 2021. Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12997–13007.
[77] Wei Zhai, Yang Cao, HaiYong Xie, and Zheng-Jun Zha. 2023. Deep Texton-Coherence Network for Camouflaged Object Detection. *IEEE Transactions on Multimedia* 25 (2023), 5155–5165.
[78] Miao Zhang, Shuang Xu, Yongri Piao, Dongxiang Shi, Shusen Lin, and Huchuan Lu. 2022. Preynet: Preying on camouflaged objects. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5323–5332.
[79] Shizhou Zhang, Dexuan Kong, Yinghui Xing, Yue Lu, Lingyan Ran, Guoqiang Liang, Hexu Wang, and Yanning Zhang. 2025. Frequency-Guided Spatial Adaptation for Camouflaged Object Detection. *IEEE Transactions on Multimedia* 27 (2025), 72–83.
[80] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. 2019. EGNet: Edge guidance network for salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8779–8788.
[81] Xiangyun Zhao, Raviteja Vemulapalli, Philip Andrew Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. 2021. Contrastive learning for label efficient semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10623–10633.
[82] Yunfei Zheng, Xiongwei Zhang, Feng Wang, Tieyong Cao, Meng Sun, and Xiaobing Wang. 2018. Detection of people with camouflage pattern via dense deconvolution network. *IEEE Signal Processing Letters* 26, 1 (2018), 29–33.
[83] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. 2022. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4504–4513.
[84] Tao Zhou, Yi Zhou, Chen Gong, Jian Yang, and Yu Zhang. 2022. Feature Aggregation and Propagation Network for Camouflaged Object Detection. *IEEE Transactions on Image Processing* 31 (2022), 7036–7047.
[85] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*. Springer, 3–11.
[86] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jing Qin. 2022. I can find you! boundary-guided separated attention network for camouflaged object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3608–3616.

Received 10 February 2025; revised 7 August 2025; accepted 9 September 2025