Boundary-Guided Camouflaged Object Detection

Yujia Sun\textsuperscript{1}, Shuo Wang\textsuperscript{2}, Chenglizhao Chen\textsuperscript{3}, Tian-Zhu Xiang\textsuperscript{4}\textsuperscript{*}

\textsuperscript{1}School of Computer Science, Inner Mongolia University, China
\textsuperscript{2}ETH Zurich, Switzerland
\textsuperscript{3}College of Computer Science and Technology, China University of Petroleum, China
\textsuperscript{4}Inception Institute of Artificial Intelligence, UAE

\texttt{tianzhu.xiang19@gmail.com}

Abstract

Camouflaged object detection (COD), segmenting objects that are elegantly blended into their surroundings, is a valuable yet challenging task. Existing deep-learning methods often fall into the difficulty of accurately identifying the camouflaged object with complete and fine object structure. To this end, in this paper, we propose a novel boundary-guided network (BGNet) for camouflaged object detection. Our method explores valuable and extra object-related edge semantics to guide representation learning of COD, which forces the model to generate features that highlight object structure, thereby promoting camouflaged object detection of accurate boundary localization. Extensive experiments on three challenging benchmark datasets demonstrate that our BGNet significantly outperforms the existing 18 state-of-the-art methods under four widely-used evaluation metrics. Our code is publicly available at: https://github.com/thograce/BGNet.

1 Introduction

Camouflage is an important defense mechanism in nature that helps certain species hide in the surroundings to protect themselves from their predators, through concealment by means of materials, coloration or illumination, or self-disguise as something else, such as imitating the appearance, colors, or patterns of the environment and disruptive coloration [Price \textit{et al.}, 2019]. This mechanism also affects human life, such as art, culture and design (e.g., camouflaged uniforms) [Stevens \textit{et al.}, 2009]. Recently, identifying a camouflaged object from its background, namely camouflaged object detection (COD), has attracted increasing research interest from computer vision community. It has promising prospects for facilitating various valuable applications in different fields, ranging from animal conservation, e.g., species discovery [Ricardo \textit{et al.}, 2012] and animal monitoring, to vision-related areas, including image synthesis [Fan \textit{et al.}, 2020a], medical image analysis [Fan \textit{et al.}, 2020b], and search-and-rescue. However, COD is a very challenging task due to the nature of camouflage, that is, the high intrinsic similarities between candidate objects and chaotic background, which make it difficult to spot camouflaged objects for humans and machines.

To tackle this issue, numerous deep learning-based methods have been proposed for camouflaged object detection and have shown great potentials. These methods can be broadly divided into three types. One is to design targeted network modules/architectures to effectively explore discriminative camouflaged-object features for COD, such as C\textsuperscript{2}FNet [Sun \textit{et al.}, 2021], UGTR [Yang \textit{et al.}, 2021]. One is to incorporate some auxiliary tasks into the joint learning/multi-task learning frameworks, such as classification task [Le \textit{et al.}, 2019], edge extraction [Zhai \textit{et al.}, 2021], salient object detection [Li \textit{et al.}, 2021], and camouflaged object ranking [Lv \textit{et al.}, 2021]. This kind of method can excavate valuable and extra cues from the shared features to significantly enhance the feature representation for COD. Another is the bio-inspired method, which mimics the behavior process of predators in nature or human visual psychological patterns to design the networks, such as SiNet [Fan \textit{et al.}, 2020a; Fan \textit{et al.}, 2021a], MirrorNet [Yan \textit{et al.}, 2021], and PFNet [Mei \textit{et al.}, 2021].

Although these recently proposed methods have made significant progress, there are still some major issues. Existing methods are often difficult to effectively and completely identify the structure and details of objects due to the edge disruption or body outline disguising, thus providing unsatisfactory

![Figure 1: Visual examples of camouflaged object detection in some challenging scenarios.](image-url)
predictions with coarse/incomplete object boundaries. As shown in Fig. 1, the ambiguities, caused by the high similarities between the butterfly head/heron legs and its background surroundings, make the features extracted by the recently state-of-the-art JCSOD [Li et al., 2021] and MGL [Zhai et al., 2021] (including single-stage version and recurrent version) models indistinguishable. Thus these models are incapable of recovering the boundary details of butterfly head and heron legs. As mentioned in [Zhao et al., 2019], the edge prior is widely used as an effective auxiliary cue, which benefits the preservation of object structure, yet has been barely studied for COD. Empirically, it is necessary to study how to enhance object-related edge visibility for facilitating the feature learning of COD. To our best knowledge, the mutual relation of annotation and ranking of camouflaged objects, [Lv et al., 2021] proposed a ranking-based testing dataset, named NC4K, which contains 4,121 images with extra localization annotation and ranking annotation.

Camouflaged object detection. In recent years, camouflaged object detection has attracted increasing attention in the computer vision community [Pang et al., 2022; Cheng et al., 2022]. Since the release of large-scale data sets (e.g., CAMO and COD10K), numerous deep learning-based camouflaged object detection models have been proposed and achieved great progress. These methods can be roughly divided into three types. The first type of method is to design advanced network modules/architectures to explore discriminative camouflaged features for COD. [Sun et al., 2021] designed an attention-induced cross-level fusion module and a dual-branch global context module to enhance feature representation. [Yang et al., 2021] incorporated the Bayesian learning into transformer-based reasoning, which can leverage both deterministic and probabilistic information for COD. The second type of method is to combine some auxiliary tasks into the joint learning frameworks to boost the performance of COD. [Le et al., 2019] proposed an anabranch network that performs an auxiliary classification task and incorporated it into mutual graph learning for COD. [Zhai et al., 2021] exploited edge extraction as an auxiliary task and incorporated it into mutual graph learning for COD. [Li et al., 2021] presented a joint salient object detection and camouflaged object detection model. [Lv et al., 2021] designed a ranking-based COD model in a joint learning framework that can mutually boost the performance of each another. The last one is the bio-inspired method, which is inspired by the behavior process of predators in nature or human visual psychological patterns. [Fan et al., 2020a] presented a search identification network to gradually locate and search for the camouflaged object, inspired by the process of wild predators discovering prey. [Mei et al., 2021] proposed a positioning and focus network by mimicking the detection and identification stages of predation.

3 Proposed Method

3.1 Overall Architecture

The overall architecture of the proposed BGNet is illustrated in Fig. 2. Specifically, we adopt Res2Net-50 [Gao et al.,
contain global location information (fmentation and localization [Zhang et al., 2017; Zhao et al., 2019] as our backbone network to extract multi-level features from the input image, i.e., $f_i$ ($i = 1, 2, \ldots, 5$). Then, an edge-aware module (EAM) is applied to excavate object-related edge semantics from the low-level features, which contain local edge details ($f_2$), and the high-level features, which contain global location information ($f_5$) under object boundary supervision. Following multiple edge-guidance feature modules (EFM) are leveraged to integrate the edge cues from EAM with multi-level backbone features ($f_2$-$f_5$) at each level to guide feature learning, which enhances boundary representation. Finally, multiple context aggregation modules (CAM) are employed to progressively aggregate multi-level fused features in a top-down manner and discover camouflaged objects. In testing, we select the prediction of the last CAM as the final result. Noted that, we do not adopt the $f_1$ backbone feature because it is too close to the input with much redundant information and a small receptive field.

### 3.2 Edge-aware Module

A good edge prior can benefit object detection in both segmentation and localization [Zhang et al., 2017; Zhao et al., 2019]. Although low-level features contain rich edge details, they also introduce many non-object edges. Thus, high-level semantic or location information is needed to facilitate the exploration of camouflaged object-related edge features. In this module, we incorporate the low-level feature ($f_2$) and the high-level feature ($f_5$) to model the object-related edge information, as shown in Fig. 3. Specifically, two $1 \times 1$ convolution layers are first used to change the channels of $f_2$ and $f_5$ to 64 ($f'_2$) and 256 ($f'_5$), respectively. Then we integrate the feature $f'_2$ and the up-sampled $f'_5$ by concatenation operation. Finally, we obtain the edge feature $f_e$ through two $3 \times 3$ con-

### 3.3 Edge-guidance Feature Module

The edge-guidance feature module (EFM) is designed to inject boundary-related edge cues into the representation learning to enhance the feature representation with object structure semantics. As is known to all, different feature channels often contain differentiated semantics. Thus, to achieve good integration and obtain a powerful representation, we introduce a local channel attention mechanism to explore cross-channel interaction and mine the critical cues between channels. As shown in Fig. 4, given the input feature $f_i$ ($i \in \{2, 3, \ldots, 5\}$) and the edge feature $f_e$, we first perform the element-wise multiplication between them with an additional skip-connection and a $3 \times 3$ convolution to obtain the initial fused features $f'_i$, which can be denoted as:

$$f'_i = F_{conv}((f_i \otimes D(f_e)) \oplus f_i),$$  \hspace{1cm} (1)$$

where $D$ denotes down-sampling and $F_{conv}$ is $3 \times 3$ convolution. $\otimes$ is element-wise multiplication and $\oplus$ is element-wise addition. To enhance feature representation, inspired by [Wang et al., 2020], we introduce local attention to explore critical feature channels. Specifically, we aggregate the convolution features ($f'_i$) using a channel-wise global average pooling (GAP). Then we obtain the corresponding channel attention (weight) by the 1D convolutions followed by a Sigmoid function. Unlike fully-connected operations, which capture dependencies across all channels but show high complexity, we explore local cross-channel interaction and learn each attention in a local manner, e.g., only consider its $k$ neighbors of every channel. After that, we multiply the channel attention with the input feature $f'_i$ and reduce the channels.
by $1 \times 1$ convolution layer to obtain the final output $f_i^a$, i.e.,
\[ f_i^a = F_{\text{conv}}(\sigma(F_{1D}^k(GAP(f_i^c))) \otimes f_i^c), \]
where $F_{\text{conv}}$ is $1 \times 1$ convolution, $F_{1D}^k$ is 1D convolution with kernel size $k$ and $\sigma$ denotes Sigmoid function. The kernel size $k$ can be set adaptively as $k = \lceil (1 + \log_2(C))/2 \rceil_{\text{odd}}$, where $\lceil \cdot \rceil_{\text{odd}}$ denotes the nearest odd number and $C$ is the channels of $f_i^c$. The kernel size is proportional to channel dimension. Obviously, the proposed attention strategy can extract multi-scale contextual features by a series of atrous convolutions, that is, to integrate the features of adjacent branches to vide cross-scale interaction into account to boost feature representation among various branches [Wu et al., 2021].

### 3.4 Context Aggregation Module

To integrate multi-level fused features for camouflaged object prediction, we design a context aggregation module (CAM) to mine contextual semantics for enhancing feature representation.

To demonstrate the effectiveness of our method, we compare it against 18 state-of-the-art methods, including 10 SOD models, i.e., PoolNet [Liu et al., 2019], EGNet [Zhao et al., 2019], SRCN [Wu et al., 2019], F3Net [Wei et al., 2020], ITSD [Zhou et al., 2020], CSNet [Gao et al., 2020], MINet [Pang et al., 2020], UCNet [Zhang et al., 2020], PraNNet [Fan et al., 2020b] and BASNet [Qin et al., 2021], and 8 COD models, i.e., SINet [Fan et al., 2020a], PFNet [Mei et al., 2021], S-MGL [Zhai et al., 2021], R-MGL [Zhai et al., 2021], LSR [Lv et al., 2021], UGTR [Yang et al., 2021], C^2FNet [Sun et al., 2021] and JCSOD [Li et al., 2021]. For fair comparison, all the predictions of these methods are either provided by the authors or produced by models retrained with open source codes.

### 4 Experiments

#### 4.1 Implementation Details

We implement our model with PyTorch and employ Res2Net-50 [Gao et al., 2019] pre-trained on ImageNet as our backbone. We resize all the input images to 416x416 and augment them by randomly horizontal flipping. During the training stage, the batch size is set to 16, and the Adam optimizer [Kingma and Ba, 2014] is adopted. The learning rate is initialized to 1e-4 and adjusted by poly strategy with the power of 0.9. Accelerated by an NVIDIA Tesla P40 GPU, the whole training takes about ~2 hours with 25 epochs.

### 4.2 Datasets

We evaluate our method on three public benchmark datasets: CAMO [Le et al., 2019], COD10K [Fan et al., 2020a] and NC4K [Lv et al., 2021]. We follow the previous works [Fan et al., 2020a], which use the training set of CAMO and COD10K as our training set, and use their testing set and NC4K as our testing sets.

### 4.3 Evaluation Metrics

We utilize four widely used metrics to evaluate our method, i.e., mean absolute error (MAE, $M$) [Perazzi et al., 2014], weighted F-measure ($F^w$) [Margolin et al., 2014], structure-measure ($S_e$) [Fan et al., 2017] and mean E-measure ($E_\alpha$) [Fan et al., 2021b].
Quantitative Evaluation. Table 1 reports the quantitative results of our method against 18 competitors on three datasets. It is obvious that our method outperforms all other models on three datasets under four evaluation metrics. Specifically, compared with the second-best JCSOD, our method increases $S_o$ by 1.93%, $E_\phi$ by 1.40% and $F_\beta^w$ by 3.55% on average. Compared with the third-best C2FNet, our method increases $S_o$ by 1.93%, $E_\phi$ by 1.41% and $F_\beta^w$ by 4.28% on average.

Table 1: Quantitative comparison with state-of-the-art methods for COD on three benchmarks using four widely used evaluation metrics ($i.e., S_o, E_\phi, F_\beta^w$, and $M$). “↑” and “↓” indicates that larger/smaller is better. Top three results are highlighted in red, green and blue.

Table 2: Quantitative evaluation for ablation studies on three datasets. The best results are highlighted in Bold. B: baseline. M: model.

4.5 Ablation Study

In order to validate the effectiveness of each key component, we design several ablation experiments and report the results in Tab. 2. For baseline model (B), we remove all the additional modules ($i.e.,$ EAM, EFM and CAM), and only retain the $1 \times 1$ convolution in four EFMs to reduce the channels of the backbone features ($f_i$, $i \in \{2,3,4,5\}$) and use the initial aggregation operation in the CAM to fuse the multi-level features in the top-down manner.

Effectiveness of CAM. From Tab. 2, compared with B model, the B+CAM model provides better performance. Especially, our module has more advantages on the metrics $F_\beta^w$ that shows 1.50% performance increases averagely.

Effectiveness of Edge Cues (EAM). To verify the effectiveness of object-related edge cues, we keep the initial fusion operation and final $1 \times 1$ convolution in EFMs, and remove the local channel attention (LCA). From Tab. 2, the model c (B+EAM+EFM w/o LCA) achieves better overall performance compared with the baseline model a, especially in terms of $F_\beta^w$ with 1.15% performance gains on average for all datasets. Thus, the edge prior extracted by EAM is beneficial to boost detection performance.

Effectiveness of EFM. Then we add LCA on model c, that is, the complete EFM, to validate effectiveness of the integra-
Figure 6: Visual comparison of the proposed model with eight state-of-the-art COD methods. Obviously, our method is capable of accurately segmenting various camouflaged objects with more clear boundaries.

Table 3: Ablation studies of different inputs of EAM on three datasets. The best results are highlighted in **Bold**.

<table>
<thead>
<tr>
<th>Method</th>
<th>CAMO-Test</th>
<th>COD10K-Test</th>
<th>NC4K</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_\alpha$</td>
<td>$E_\phi$</td>
<td>$F_\beta$</td>
</tr>
<tr>
<td>$f_1 + f_5$</td>
<td>0.807</td>
<td>0.865</td>
<td>0.737</td>
</tr>
<tr>
<td>$f_3 + f_5$</td>
<td>0.805</td>
<td>0.863</td>
<td>0.738</td>
</tr>
<tr>
<td>$f_2 + f_5$ (Ours)</td>
<td><strong>0.812</strong></td>
<td><strong>0.870</strong></td>
<td><strong>0.749</strong></td>
</tr>
</tbody>
</table>

Figure 7: Examples of object-related edge exploration in EAM.

5 Conclusion

In this paper, we resort to edge priors to assist recovering object structure and boost the performance of camouflaged object detection. We propose a simple yet effective boundary-guided network (BGNet), which contains edge-aware module, edge-guidance feature module, and context aggregation module, to explore object-related edge semantics to guide and enhance representation learning for COD. By adopting edge cues, our BGNet provides accurate camouflaged object predictions with complete and fine object structure and boundaries. Extensive experiments show that our method outperforms existing state-of-the-art methods on three benchmarks.

References

[Cheng et al., 2022] X Cheng, H Xiong, D-P Fan, Y Zhong, and et al. Implicit motion handling for video camouflaged


