Boundary-Guided Camouflaged Object Detection

Yujia Sun¹, Shuo Wang², Chenglizhao Chen³, Tian-Zhu Xiang⁴*

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

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

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 C²FNet [Sun et al., 2021], UGTR [Yang et al., 2021]. One is to incorporate some auxiliary tasks into the joint learning/multi-task learning frameworks, such as classification task [Le et al., 2019], edge extraction [Zhai et al., 2021], salient object detection [Li et al., 2021], and camouflaged object ranking [Lv 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 et al., 2020a; Fan et al., 2021a], MirrorNet [Yan et al., 2021], and PFNet [Mei 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
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 graph learning model (MGL) [Zhai et al., 2021] is the first to explicitly exploit the edge information to improve the performance of COD. However, MGL encodes edge features into graph convolutional networks along with object features and boosts feature representation through graph interaction modules. It can be seen that MGL is a sophisticated model, which inevitably increases model complexity and suffers from the heavy computational burden. In addition, it is noted that, despite the introduction of edge cues, MGL still loses some fine boundary-related details and thus weakens the performance of COD. As shown in the 2nd row of Fig. 1, the MGL models lose many details about heron legs and introduces some obvious background noises in prediction results.

To this end, in this paper, we propose a novel boundary-guided network (BGNet), which explicitly employs edge semantic to enhance the performance of camouflaged object detection. Firstly, we design a simple yet effective edge-aware module (EAM), which integrates the low-level local edge information and high-level global location information to explore edge semantics related to object boundaries under explicitly boundary supervision. Then, the edge-guidance feature module (EFM) is introduced to incorporate the edge features with the camouflaged object features at various layers to guide the representation learning of COD. The EFM module can enforce the network to pay more attention to the object structure and details. After that, the multi-level fused features are aggregated gradually from top to bottom to predict camouflaged objects. To enhance the feature representation, we build a context aggregation module (CAM), which mines and aggregates multi-scale contextual semantics by a series of atrous convolutions to produce features with stronger and more effective representations. Benefiting from the well-designed modules, the proposed BGNet predicts camouflaged objects with fine object structure and boundaries. Note that, compared with MGL, we design a simpler but more effective edge extraction module to excavate accurate object boundary semantics, and then guide the feature representation learning of camouflaged objects by means of the proposed EFM and CAM. Besides, our method achieves more accurate object localization and stronger preservation of object structure. To sum up, our main contributions are as follows:

- We carefully design the edge-guidance feature module (EFM) and context aggregation module (CAM) to enhance boundary semantics and explore valuable and powerful feature representation for COD.

2 Related Work

Datasets. To boost deep learning-based camouflaged object detection, some annotation datasets are proposed. [Le et al., 2019] constructed the first camouflaged object dataset, namely CAMO, including 1,250 camouflaged images covering eight categories. [Fan et al., 2020a] collected a large-scale challenging dataset, called COD10K, which contains 10,000 images covering 78 camouflaged object categories, with high-quality and hierarchical annotations. Recently, to support localization 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 anabranche network that performs an auxiliary classification network to help camouflaged object segmentation. [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 network to enhance the detection ability of both tasks. To segment and rank camouflaged objects, [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.,
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. 8. Specifically, two 1×1 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 3×3 convolutions and one 1×1 convolution layer followed by the Sigmoid function. EAM is a simple yet effective module to extract specific edge features. As shown in Fig. 7, EAM perfectly learns the object boundary-related edge semantics.

### 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 3×3 convolution to obtain the initial fused features \(f''_i\), which can be denoted as:

\[
f''_i = F_{\text{conv}}((f'_i \otimes D(f_e)) \oplus f_i),
\]

where \(D\) denotes down-sampling and \(F_{\text{conv}}\) is 3×3 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.
where $F$ with kernel size $\sigma$ and $\eta$ denoted as an example, we first upsample $f^3_{\text{conv}}$ and concatenate them followed by a $1 \times 1$ convolution layer to obtain the initial aggregated feature $f_m$. Next, we divide $f_m$ into four feature maps $f_m^{1}, f_m^{2}, f_m^{3}, f_m^{4}$ along channel dimension, and then perform cross-scale interaction learning, that is, to integrate the features of adjacent branches to extract multi-scale contextual features by a series of atrous convolutions. It can be formulated as:

$$f_m^{j} = F_{\text{conv}}(\sigma(F_{\text{1D}}^{k}(GAP(f_i^{j})))) \otimes f_i^{j}, \quad j \in \{1,2,3,4\},$$

where $F_{\text{conv}}$ is $1 \times 1$ convolution, $F_{\text{1D}}^{k}$ is 1D convolution with kernel size $k$ and $\sigma$ denotes Sigmoid function. The kernel size $k$ can be set adaptively as $k = \lfloor(1 + \log_2(C)) / 2 \rfloor_{\text{odd}}$, where $\lfloor x \rfloor_{\text{odd}}$ denotes the nearest odd number and $C$ is the channel number of feature $f_i^{j}$. The kernel size is proportional to channel dimension. Besides, for $f_i=1$ convolution, a residual block is used to feed-forward the original feature $f_i^{j}$. In our experiments, we set $n_j = \frac{1}{2}$, $i = 1,2,3,4$.

Then, we concatenate these four multi-scale features $f_i^{j}$ followed by a $1 \times 1$ convolution, a residual connection and a $3 \times 3$ convolution, which can be denoted as:

$$f_i^{c} = F_{\text{conv}}(\sigma(F_{\text{conv}}(f_m^{j}))) \otimes f_m^{j},$$

where $\otimes$ indicates a channel number of feature $f_i^{c}$, we can obtain the prediction $P_i (i \in \{2,3,4\}$ of camouflaged objects.

3.5 Loss Function

Our model has two kinds of supervisions: camouflaged object mask ($G_o$) and camouflaged object edge ($G_e$). For mask supervision, we employ the weighted binary cross-entropy loss ($L_{\text{BCE}}$) and weighted IOU loss ($L_{\text{IOU}}$) [Wei et al., 2020], which pay more attention to hard pixels instead of assigning all pixels equal weights. For edge supervision, we adopt the dice loss ($L_{\text{dice}}$) [Xie et al., 2020] to deal with the strong imbalance between positive and negative samples. Noted that the mask supervision is conducted on three camouflaged object predictions ($P_i, i \in \{2,3,4\}$) from CAM. Thus, the total loss is defined as:

$$L_{\text{total}} = \sum_{i=2}^{4}(\lambda L_{\text{BCE}}(P_i, G_o) + \lambda L_{\text{IOU}}(P_i, G_o)) + \lambda L_{\text{dice}}(P_e, G_e),$$

where $\lambda$ is a trade-off parameter and set $\lambda = 3$ in our experiments, $P_e$ is the prediction of camouflaged object edges.

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 416 x 416 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, $\lambda$), weighted F-measure ($F_{\beta}$) [Margolin et al., 2014], structure-measure ($S_o$) [Fan et al., 2017] and mean E-measure ($E_o$) [Fan et al., 2021b].

4.4 Comparison with State-of-the-arts

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], EGRNet [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], PraNet [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], C2FNet [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.
Table 1: Quantitative comparison with state-of-the-art methods for COD on three benchmarks using four widely used evaluation metrics (i.e., $S_o$, $E_o$, $F_p^\beta$, and $M$). “↑” and “↓” indicates that larger/smaller is better. Top three results are highlighted in red, green and blue.

| Method     | Pub./Year | CAMO-Test | COD10K-Test | NC4K |
|------------|-----------|-----------|-------------|------|
|            | $S_o$ ↑ | $E_o$ ↑ | $F_p^\beta$ ↑ | $M$ ↓ | $S_o$ ↑ | $E_o$ ↑ | $F_p^\beta$ ↑ | $M$ ↓ | $S_o$ ↑ | $E_o$ ↑ | $F_p^\beta$ ↑ | $M$ ↓ |
| PoolNet    | CVPR’19   | 0.730     | 0.746       | 0.575     | 0.105 | 0.740     | 0.776       | 0.506     | 0.056 | 0.785     | 0.814       | 0.635     | 0.073 |
| EGN        | ICCV’19   | 0.732     | 0.800       | 0.604     | 0.109 | 0.736     | 0.810       | 0.517     | 0.061 | 0.777     | 0.841       | 0.639     | 0.075 |
| SRCN       | ICCV’19   | 0.779     | 0.797       | 0.643     | 0.090 | 0.789     | 0.817       | 0.575     | 0.047 | 0.830     | 0.854       | 0.698     | 0.059 |
| F²Net      | AAAI’20   | 0.711     | 0.741       | 0.564     | 0.109 | 0.739     | 0.795       | 0.544     | 0.051 | 0.780     | 0.824       | 0.656     | 0.070 |
| ITSD       | CVPR’20   | 0.750     | 0.779       | 0.610     | 0.102 | 0.767     | 0.808       | 0.557     | 0.051 | 0.811     | 0.844       | 0.680     | 0.064 |
| CSNet      | ECCV’20   | 0.771     | 0.809       | 0.569     | 0.092 | 0.778     | 0.809       | 0.569     | 0.047 | 0.750     | 0.773       | 0.603     | 0.088 |
| MNet       | CVPR’20   | 0.748     | 0.791       | 0.637     | 0.090 | 0.770     | 0.832       | 0.608     | 0.042 | 0.812     | 0.862       | 0.720     | 0.056 |
| UCNet      | CVPR’20   | 0.739     | 0.787       | 0.640     | 0.094 | 0.776     | 0.857       | 0.633     | 0.042 | 0.811     | 0.871       | 0.729     | 0.055 |
| PraNet     | MICCAI’20 | 0.769     | 0.825       | 0.663     | 0.094 | 0.789     | 0.861       | 0.629     | 0.045 | 0.822     | 0.876       | 0.724     | 0.059 |
| BASNet     | arxiv’21  | 0.749     | 0.796       | 0.646     | 0.096 | 0.802     | 0.855       | 0.677     | 0.038 | 0.817     | 0.859       | 0.732     | 0.058 |
| SINet      | CVPR’20   | 0.745     | 0.804       | 0.644     | 0.092 | 0.776     | 0.864       | 0.631     | 0.043 | 0.808     | 0.871       | 0.723     | 0.058 |
| FFNet      | CVPR’21   | 0.782     | 0.841       | 0.695     | 0.085 | 0.800     | 0.877       | 0.660     | 0.040 | 0.829     | 0.887       | 0.745     | 0.053 |
| S-MGL      | CVPR’21   | 0.772     | 0.806       | 0.664     | 0.089 | 0.811     | 0.844       | 0.654     | 0.037 | 0.829     | 0.862       | 0.731     | 0.055 |
| R-MGL      | CVPR’21   | 0.775     | 0.812       | 0.673     | 0.088 | 0.814     | 0.851       | 0.666     | 0.035 | 0.833     | 0.867       | 0.739     | 0.053 |
| UGTR       | ICCV’21   | 0.784     | 0.821       | 0.683     | 0.086 | 0.817     | 0.852       | 0.665     | 0.036 | 0.839     | 0.874       | 0.746     | 0.052 |
| LSR        | CVPR’21   | 0.787     | 0.838       | 0.696     | 0.080 | 0.804     | 0.880       | 0.673     | 0.037 | 0.840     | 0.895       | 0.766     | 0.048 |
| C²FNet     | IJCAI’21  | 0.796     | 0.854       | 0.719     | 0.080 | 0.813     | 0.890       | 0.686     | 0.036 | 0.838     | 0.897       | 0.762     | 0.049 |
| JCSCD      | CVPR’20   | 0.800     | 0.859       | 0.728     | 0.073 | 0.809     | 0.884       | 0.684     | 0.035 | 0.841     | 0.898       | 0.771     | 0.047 |
| BGNet (Ours) | IJCAI’22 | 0.812     | 0.870       | 0.749     | 0.073 | 0.831     | 0.901       | 0.722     | 0.033 | 0.851     | 0.907       | 0.788     | 0.044 |

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 1×1 convolution in four EFMs to reduce the channels of the backbone features ($f_i, i = \{2, 3, 4, 5\}$) 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 $F_p^\beta$ 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 of the auxiliary edge detection network based on a complex graph model, it still loses many structural details, resulting in poor boundary localization in prediction. It is proven that our method achieves superior performance in object-related edge information mining and camouflaged object prediction.

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 1×1 convolution in four EFMs to reduce the channels of the backbone features ($f_i, i = \{2, 3, 4, 5\}$) 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 $F_p^\beta$ 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 of the auxiliary edge detection network based on a complex graph model, it still loses many structural details, resulting in poor boundary localization in prediction. It is proven that our method achieves superior performance in object-related edge information mining and camouflaged object prediction.
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.

Input of EAM. We also test the effectiveness of different inputs for EAM, e.g., $f_1$, $f_2$ and $f_3$ used to explore edges with $f_5$ to help locate object-related edges. As shown in Tab. 3, the combination of $f_2 + f_5$ obtains the best performance for camouflaged object detection.

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
object detection. In CVPR, 2022.

[Fan et al., 2017] D-P Fan, M-M Cheng, Y Liu, T Li, and A Borji. Structure-measure: A new way to evaluate foreground maps. In ICCV, 2017.

[Fan et al., 2020a] D-P Fan, G-P Ji, G Sun, M-M Cheng, J Shen, and L Shao. Camouflaged object detection. In CVPR, 2020.

[Fan et al., 2020b] D-P Fan, G-P Ji, T Zhou, G Chen, H-Z Fu, J-B Shen, and L Shao. Pranet: Parallel reverse attention network for polyp segmentation. In MICCAI, 2020.

[Fan et al., 2020c] D-P Fan, Y Zhao, A Borji, and et al. Bbsnet:Rgb-d salient object detection with a bifurcated backbone strategy network. In ECCV, 2020.

[Fan et al., 2021a] D-P Fan, G-P Ji, M-M Cheng, and L Shao. Concealed object detection. IEEE TPAMI, 2021.

[Fan et al., 2021b] D-P Fan, G-P Ji, X Qin, and M-M Cheng. Cognitive vision inspired object segmentation metric and loss function. In SSI, 2021.

[Gao et al., 2019] S Gao, M Cheng, K Zhao, X Zhang, M Yang, and P Torr. Res2net: A new multi-scale backbone architecture. IEEE TPAMI, 2019.

[Gao et al., 2020] S Gao, Y Tan, M-M Cheng, C Lu, Y Chen, and S Yan. Highly efficient salient object detection with 100k parameters. In ECCV, 2020.

[Kingma and Ba, 2014] D P Kingma and J Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[Le et al., 2019] T-N Le, T V Nguyen, Z Nie, M-T Tran, and A Sugimoto. Anabranch network for camouflaged object segmentation. CVIU, 2019.

[Li et al., 2021] A Li, J Zhang, Y Lv, B Liu, T Zhang, and Y Dai. Uncertainty-aware joint salient object and camouflaged object detection. In CVPR, 2021.

[Liu et al., 2019] J-J Liu, Q Hou, M-M Cheng, J Feng, and J Jiang. A simple pooling-based design for real-time salient object detection. In CVPR, 2019.

[Lv et al., 2021] Y Lv, J Zhang, Y Dai, A Li, B Liu, N Barnes, and D Fan. Simultaneously localize, segment and rank the camouflaged objects. In CVPR, 2021.

[Margolin et al., 2014] R Margolin, L Zelnik-Manor, and A Tal. How to evaluate foreground maps? In CVPR, 2014.

[Mei et al., 2021] H Mei, G Ji, Z Wei, X Wei, X Yang, and D Fan. Camouflaged object segmentation with distraction mining. In CVPR, 2021.

[Pang et al., 2020] Y Pang, X Zhao, L Zhang, and H Lu. Multi-scale interactive network for salient object detection. In CVPR, 2020.

[Pang et al., 2022] Y Pang, X Zhao, T-Z Xiang, L Zhang, and H Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In CVPR, 2022.

[Perazzi et al., 2012] F Perazzi, P Krähenbühl, Y Pritch, and A Hornung. Saliency filters: Contrast based filtering for salient region detection. In CVPR, 2012.

[Price et al., 2019] N Price, S Green, J Troscianko, T Tregenza, and M Stevens. Background matching and disruptive coloration as habitat-specific strategies for camouflage. Scientific reports, 9(1):1–10, 2019.

[Qin et al., 2021] X-B Qin, D-P Fan, C-Y Huang, and et al. Boundary-aware segmentation network for mobile and web applications. arXiv preprint arXiv:2101.04704, 2021.

[Ricardo et al., 2012] P F Ricardo, X Delclòs, E Peñalver, and et al. Early evolution and ecology of camouflage in insects. PNAS, 109(52):21414–21419, 2012.

[Steven et al., 2009] Martin Stevens, Sami Merilaita, et al. Animal camouflage: current issues and new perspectives. RSTB, 364(1516):423–427, 2009.

[Sun et al., 2021] Y Sun, G Chen, T Zhou, Y Zhang, and N Liu. Context-aware cross-level fusion network for camouflaged object detection. In IJCAI, 2021.

[Wang et al., 2020] Q Wang, B Wu, P Zhu, P Li, W Zuo, and Q Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In CVPR, 2020.

[Wei et al., 2020] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: Fusion, feedback and focus for salient object detection. In AAAI, 2020.

[Wu et al., 2019] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In CVPR, 2019.

[Wu et al., 2020] Y Wu, Y Liu, L Zhang, and et al. EDN: Salient object detection via extremely-downsampled network. arXiv preprint arXiv:2012.13093, 2020.

[Xie et al., 2020] Enze Xie, Wenjia Wang, Wenhai Wang, Mingyu Ding, Chunhua Shen, and Ping Luo. Segmenting transparent objects in the wild. In ECCV, 2020.

[Yan et al., 2021] J Yan, T-N Le, K-D Nguyen, M-T Tran, and T-T Do. Mirrornet: Bio-inspired camouflaged object segmentation. IEEE Access, 9:43290–43300, 2021.

[Yang et al., 2021] F Yang, Q Zhai, X Li, R Huang, A Luo, H Cheng, and D Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In ICCV, 2021.

[Zhai et al., 2021] Q Zhai, X Li, F Yang, C Chen, H Cheng, and D Fan. Mutual graph learning for camouflaged object detection. In CVPR, 2021.

[Zhang et al., 2017] P Zhang, D Wang, H Lu, H Wang, and X Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In ICCV, 2017.

[Zhang et al., 2020] J Zhang, D-P Fan, Y Dai, S Anwar, F S Saleh, T Zhang, and N Barnes. Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In CVPR, 2020.

[Zhao et al., 2019] J-X Zhao, J-J Liu, D-P Fan, Y Cao, J Yang, and M-M Cheng. Egnet: Edge guidance network for salient object detection. In CVPR, 2019.

[Zhou et al., 2020] H Zhou, X Xie, J Lai, Z Chen, and L Yang. Interactive two-stream decoder for accurate and fast saliency detection. In CVPR, 2020.
6 Appendix

6.1 Model complexity

Our BGNet is a multi-task learning model for COD, so we report the comparison of existing COD methods based on multi-task learning (i.e., S-MGL, R-MGL, JCSOD, LSR) with ours in terms of number of parameters, FLOPs and FPS in Tab. 4. All evaluations follow the inference settings in the corresponding papers. It can be seen that our method achieves a good balance of these three metrics.

| Method | Input Size | Params. | FLOPs | FPS |
|--------|------------|---------|-------|-----|
| S-MGL  | 473×473    | 63.60M  | 236.60G| 12  |
| R-MGL  | 473×473    | 67.64M  | 249.89G| 8   |
| JCSOD  | 352×352    | 121.63M | 25.20G | 43  |
| LSR    | 352×352    | 57.90M  | 25.21G | 33  |
| BGNet  | 416×416    | 79.85M  | 58.45G | 35  |

Table 4: Comparison of model complexity.

6.2 Experimental Settings for Ablation Study

Figure 8 shows the specific experimental settings for model (b) (B+EAM+EFM w/o LCA) and model (c) (B+EAM+EFM) in Tab. 2. Comparing the results of model (b) and model (c), that is, B+EAM+EFM (w/o LCA) and B+EAM+EFM(w/ LCA), it can be found that the boundary cues are still effective even if they are fused in a simple way, while local attention is the icing on the cake.

6.3 Parameter λ

Our BGNet has three mask predictions and one boundary prediction. Intuitively, we set λ to 3 to balance these two types of losses in the total loss. We also study the different parameter settings (i.e., λ = {1, 2, 3, 4, 5}) in Tab. 5. Four metrics (i.e., $E_\phi$, $F^w_\beta$, $S_\alpha$, $M$) are adopted for performance evaluation. From these results, we can see that λ = 3 achieves the best performance compared to other settings.

| λ   | $E_\phi$ | $F^w_\beta$ | $S_\alpha$ | $M$ |
|-----|----------|--------------|------------|-----|
| λ = 1 | 0.862   | 0.743        | 0.809      | 0.073 |
| λ = 2 | 0.865   | 0.742        | 0.806      | 0.073 |
| λ = 3 | **0.869** | **0.748**   | **0.811**  | **0.073** |
| λ = 4 | 0.863   | 0.736        | 0.803      | **0.072** |
| λ = 5 | 0.867   | 0.739        | 0.806      | 0.074 |

Table 5: Quantitative results for different setting of λ on CAMO.

6.4 Relationship between modules

Our BGNet is a unified whole with serially interdependent components: EAM, EFM, and CAM, which complement each other. The former two explore boundary cues and guide the feature learning, and then the latter further enhances critical feature representation. As shown in Tab. 2 of the paper, the contributions of EAM, EFM and CAM are 1.22%, 0.65%, 1.94% on average respectively in terms of $F^w_\beta$. Thus, we can see that CAM and EAM (boundary cues) contribute more than EFM.

![Illustration of the edge-guidance feature module (EFM) with and without local channel attention (LCA) mechanism.](image-url)