Locate, Refine and Restore: A Progressive Enhancement Network for Camouflaged Object Detection

Xiaofei Li[1] , **Jiaxin Yang**[1] , **Shuohao Li**[1] , **Jun Lei**[1] , **Jun Zhang**[1*] and **Dong Chen**[2]
[1]Laboratory for Big Data and Decision, National University of Defense Technology, China
[2]Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, China
{xf, lishuohao, leijun1987, zhangjun1975, dongchen}@nudt.edu.cn, yangjiaxin0313@163.com

Abstract

Camouflaged Object Detection (COD) aims to segment objects that blend in with their surroundings. Most existing methods mainly tackle this issue by a single-stage framework, which tends to degrade performance in the face of small objects, low-contrast objects and objects with diverse appearances. In this paper, we propose a novel Progressive Enhancement Network (**PENet**) for COD by imitating the human visual detection system, which follows a three-stage detection process: locate objects, refine textures and restore boundary. Specifically, our PENet contains three key modules, i.e., the object location module (OLM), the group attention module (GAM) and the context feature restoration module (CFRM). The OLM is designed to position the object globally, the GAM is developed to refine both high-level semantic and low-level texture feature representation, and the CFRM is leveraged to effectively aggregate multi-level features for progressively restoring the clear boundary. Extensive results demonstrate that our PENet significantly outperforms 32 state-of-the-art methods on four widely used benchmark datasets.

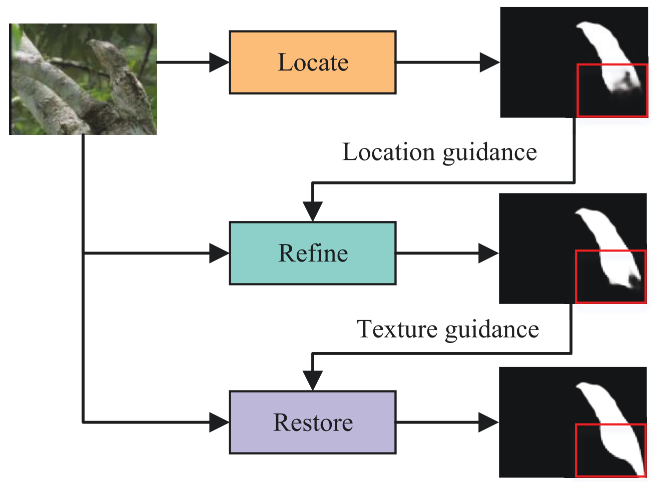

Figure 1: Illustration of our proposed progressive enhancement network (PENet). Inspired by the behavior of humans when observing vague images, our PENet is designed with three steps (i.e., locate objects, refine textures and restore boundary) to progressively improve the performance of COD.

1 Introduction

Camouflage is a widespread biological phenomenon in nature that helps certain organisms hide in the surroundings to protect themselves from predators [Cuthill *et al.*, 2005]. In practice, camouflaged objects usually conceal themselves by imitating the appearance, colors, or patterns of the environment and the disruptive coloration [Price *et al.*, 2019], making them difficult to be found. Based on this strategy, human beings began to study bionic disguise things according to their own ideas for the purpose of camouflage. For example, in military combat soldiers achieve stealth by wearing camouflage clothing made of special materials, hunters use disguised sounds in the forest to trap animals farmers build scarecrows in the fields to repel birds. Recently, camouflaged object detection (COD) has attracted an increasing research interest from the computer vision community, and a lot of potential applications have been developed in different fields, such as medical diagnosis (e.g., polyp segmentation, lung infection segmentation), agriculture (e.g., locust detection to prevent invasion), industry (surface defect detection), security and surveillance (e.g., search and rescue work), animal conservation (e.g., species discovery), and art (e.g., recreational art).

However, COD is an extremely challenging task due to the camouflaged objects making themselves "perfectly" assimilate into their surroundings by means of the materials, coloration or illumination. As shown in Fig.1, due to low contrast and the appearance of expressive diversity, the texture of the object is similar to the surroundings, it is very challenging to discover it. To address this problem, early traditional camouflaged object detectors attempt to extract discriminative hand-crafted low-level features rely on the manual visual feature, such as color, 3D convexity and appearance texture. In recent years, numerous algorithms for COD based on deep neural networks have been proposed, which can si-

*Corresponding Author

multaneously extract low-level texture features and high-level semantic information. Despite good performance achievements, there is still a large place for improvement. First, the idea of most existing approaches is to use ASPP [Chen *et al.*, 2017] modules in a single-stage manner to extract the context and then decode the segmentation results using some simple fusion methods. This method does not take into account the localization of the camouflaged object, so there is often inaccuracy in segmenting the object. Second, although there are some two-stage approaches, they usually consider only the predicted camouflaged objects from coarse to fine, without considering also the object positioning, texture enhancement and boundary restoration.

Previous biological studies [Hall *et al.*, 2013] have shown that the human eye will first locate the general position of an object when viewing it, then focus on the main area of the object to find details, and then further fuse and refine the boundary until the object is completely detected from the background. Inspired by this human behavior and considering the shortcomings of these methods, we aim to address the COD issue: *How to accurately locate the object and refine the textures to restore clear boundary under complex scenes?*

To this end, we develop a novel bio-inspired framework, termed progressive enhancement network (PENet), which significantly improves the existing camouflaged object segmentation performance. Our PENet consists of three key modules, i.e., the object location module (OLM), the group attention module (GAM) and the context feature restoration module (CFRM), to accurately locate the camouflaged object, refine the textures and restore the boundary in a progressively enhanced manner, respectively. Specifically, the OLM consists of a global attention component and a local attention component to mimic the human detection process by first locating the target globally. The GAM is designed as a group attention module that contains channel attention and spatial attention to focus on detailed features at different scales. The CFRM uses attention-guided fusion of texture features between different layers to achieve the boundary-reduction.

In summary, the main contributions of this paper can be summarized as follows:

- We present a new bio-inspired framework called progressive enhancement network (PENet) for COD, which greatly improves the performance of COD in a progressively enhanced manner to locate objects, enhance texture features and restore boundary.
- We propose an object location module (OLM) to infer the initial position of the camouflaged objects, which can effectively extract global information and local features so that the location of the camouflaged object can be accurately determined. We also design a group attention module (GAM) to refine textures, and a context feature restoration module (CFRM) to restore clear boundary.
- Extensive experiments on four benchmark datasets demonstrate that our PENet achieves the state-of-the-art performance of COD. Qualitative and quantitative results demonstrate the effectiveness of our method.

2 Related Work

2.1 Camouflaged Object Detection

Different from salient object detection that aims to detect and segments the most compelling objects in the image, the purpose of camouflaged object detection is to find objects that closely resemble their surroundings. [Le *et al.*, 2019] proposed an anabranch network, which leverages both classification and segmentation tasks. [Fan *et al.*, 2020a] proposed a Search Identification Network to address this challenge by first roughly searching for camouflaged objects, and then segmenting the objects by a recognition module. [Sun *et al.*, 2021] proposed C^2F-Net for COD, which considers global contextual information to integrate multi-level features. [Jia *et al.*, 2022] proposed an iterative refinement framework, coined SegMaR, which integrates Segment, Magnify and Reiterate in a multi-stage detection fashion.

2.2 Object Location

Accurate object localization is important for computer vision tasks, and it often affects the performance of these tasks. In order to solve the co-localization problem, [Gokberk Cinbis *et al.*, 2014] proposed a multiplicative multi-instance learning procedure to iteratively train the detector, which prevents training from prematurely locking onto erroneous object locations. [Bazzani *et al.*, 2016] proposed a self-taught object localization method that localizes objects by identifying the regions causing the maximal activations. [Zhang *et al.*, 2018a] proposed an Adversarial Complementary Learning approach for discovering entire objects of interest by two adversarial classifiers.

2.3 Multi-Scale Feature Refinement

As the CNNs become deeper, the detailed features may be diluted. To make more efficient use of texture features, one solution is to aggregate multi-scale information. [Chen *et al.*, 2017] proposed a spatial pyramid pooling to robustly segment objects at multiple scales by an incoming convolutional feature layer with filters at multiple sampling rates and effective fields-of-view. [Pang *et al.*, 2020] presented a multi-scale interactive network for salient object detection, which embeds a self-interaction module in each decoder unit in order to obtain more effective multi-scale features from the integrated features. [Zhu *et al.*, 2021] proposed an interactive guidance framework to interactively refine multi-level features of camouflaged object detection and texture detection.

2.4 Context-Aware Feature Learning

The contextual information plays a crucial role in enhancing feature representation for many computer vision tasks. [Hu *et al.*, 2018] proposed a network for shadow detection by analyzing spatial context in a direction-aware manner. [Chen *et al.*, 2020] used some progressive context-aware Feature Interweaved Aggregation (FIA) modules to integrate low-level appearance features, high-level semantic features and global contextual features. [Mei *et al.*, 2020] explored abundant contextual cues with a large-field contextual feature integration (LCFI) module for robust glass detection. [Dai *et al.*, 2021] proposed a multi-scale channel attention module to fuse features given at different scales.

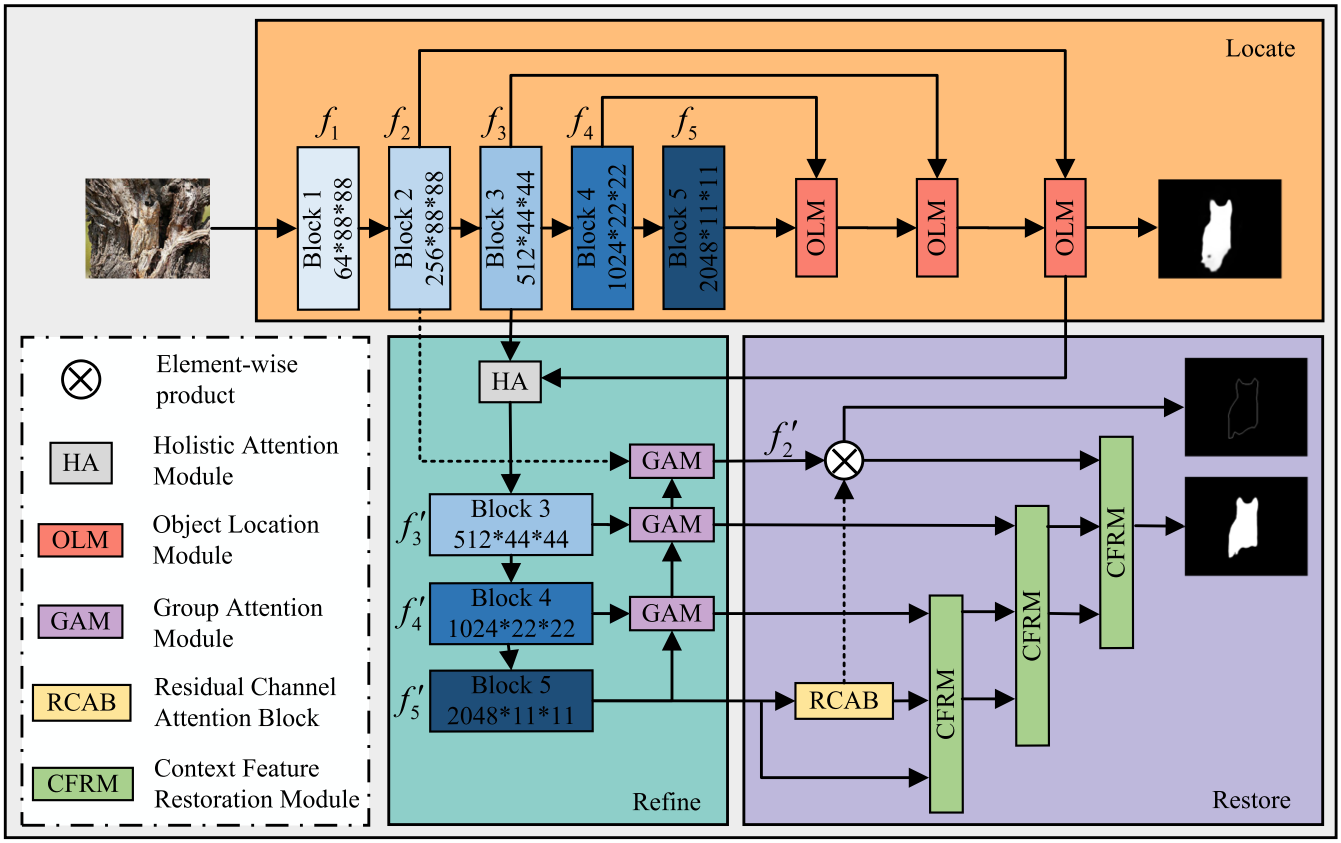

Figure 2: The overview of our proposed progressive enhancement network (PENet),which consists of three key modules, i.e., object location module (OLM), group attention module (GAM) and context feature restoration module (CFRM). See Sec. 3 for details.

3 Proposed Method

3.1 Overall Architecture

The overall framework of our PENet is shown in Fig.2. Specifically, for an input RGB image I with size $H \times W$, we adopt Res2Net-50 [Gao *et al.*, 2019] as the backbone to extract its multi-level features from the input image, denoted as $f_i(i = 1, 2, ..., 5)$. Then, we use three object location modules (OLMs) to locate the potential camouflaged objects. Next, we feed these features with their initial position to three convolutional blocks and group attention modules (GAMs) to refine the details. Finally, we leverage three context feature restoration modules (CFRMs) to restore clear and complete objects.

3.2 Initial Object Location

In the locating stage, we design an object location module (OLM) to initially locate the potential location of the camouflaged object. As shown in Fig.3, it consists of a global block and a local block. The global block is implemented in a non-local way to capture long-range dependencies for enhancing the contextual semantic representation from a global perspective. In contrast, the local block is designed to extract the local information by using several convolutional layers. These two blocks explore potential object regions in a complementary way.

Specifically, we first leverage the receptive fields block (RFB) [Liu *et al.*, 2018] structure to enlarge the receptive field, obtain four feature maps B, C, D and E through four 1×1 convolutional layers, where $\{B, C, D, E\} \in \mathbb{R}^{C\times H\times W}$, then we reshape them separately for $\mathbb{R}^{C\times N}$. Next, we multiply the transpose of B by the C matrix, and perform a softmax layer to calculate the global spatial attention maps $gsa\in \mathbb{R}^{N\times N}$. Consequently, we use matrix multiplication operation to get the semantic-enhanced global feature $f_g \in \mathbb{R}^{C\times H\times W}$. The process can be depicted as follows:

$$gsa_{ij} = \frac{exp(B_i \cdot C_j)}{\sum_{i=1}^{N} exp(B_i \cdot C_j)} \tag{1}$$

$$f_g^i = \eta \sum_{j=1}^{N} (gsa_{ij} \cdot D_j) + f_i \tag{2}$$

where ga_{ij} denotes the j^{th} position's impact on the i^{th} position. η is initialized as 0 and gradually learns more weight.

In the local block, we adopt a set of 1×1 convolutional layers, BatchNorm2d and ReLU to obtain the local features $f_l \in \mathbb{R}^{C\times H\times W}$. Then, we perform the element-wise addition operation on global feature f_g and local features f_l to obtain the aggregated features f_a. This process can be described as follows:

$$f_l = ReLU(Conv_{1\times1}(ReLU(BN(Conv_{1\times1}(E))))) \tag{3}$$

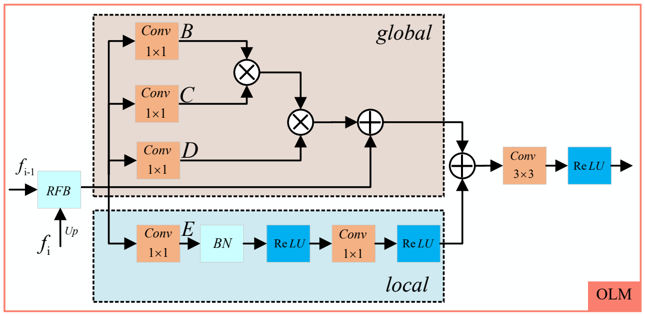

Figure 3: Detailed structure of the object location module (OLM).

$$f_a = ReLU(Conv_{3\times3}(f_g + f_l)) \tag{4}$$

where $Conv_{1\times1}$ indicates 1×1 convolutional layers, BN means the BatchNorm2d and $Conv_{3\times3}$ indicates 3×3 convolutional layers.

3.3 Texture Detail Refinement

Since the obtained initial prediction in the localization stage is coarse and contains irrelevant noise, we need to further refine the texture features. In the refinement stage, we develop a group attention pyramid network to refine the details for more effective feature representation. It contains three convolutional blocks and three group attention modules (GAM).

Specifically, as shown in Fig.2, we first use a holistic attention (HA) module [Wu *et al.*, 2019a] to merge the coarse prediction and feature maps to highlight the whole object region. Then, we use three convolutional blocks (i.e., f_3', f_4' and f_5') to get the the bottom-up features, and then construct a pyramid network from them and the features of f_2. In each level of the top-down pathway of the pyramid network, we design a group attention module (GAM) to mining multi-scale features. As shown in Fig 4, we first use the RFB structure to enlarge the receptive field, then the global features $f_g \in \mathbb{R}^{C\times H\times W}$ and the attention features $f_a \in \mathbb{R}^{C\times H\times W}$ are splited into M fixed groups along the channel dimension. After that, the splited features $\{f_g\}_{m=1}^{M} \in \mathbb{R}^{C/M\times H\times W}$ and $\{f_a\}_{m=1}^{M} \in \mathbb{R}^{C/M\times H\times W}$ are obtained. Consequently, we can get the regrouped features $f \in \mathbb{R}^{C\times H\times W}$ by a concatenation operation:

$$f = Concat(f_g^1, f_a^1, ..., f_g^m, f_a^m, ..., f_g^M, f_a^M) \tag{5}$$

where $Concat()$ denotes the concatenation operation.

To enhance the texture features, we apply spatial attention and channel attention to the regrouped features separately. For the spatial attention, we use a deconvolution layer with one output channel and a 3×3 kernel size to extract spatial information, and then we use a sigmoid function to normalize it to the range of (0,1). The process can be expressed as follows:

$$SA = \sigma(Deconv(f)) \tag{6}$$

where σ represents the sigmoid function. $Deconv$ refers to the 3×3 deconvolution layer. SA represents the spatial attention feature.

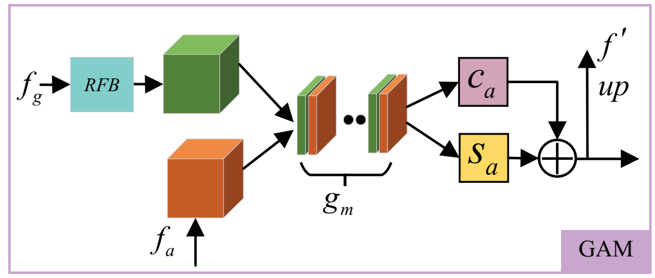

Figure 4: Detailed structure of the group attention module (GAM).

In the channel attention, we use a global average pooling (GAP) and two 1×1 convolutional layers to reduce dimension. Then a ReLU and a sigmoid function is applied to gained the channel attention:

$$CA = \sigma(Conv_{1\times1}(ReLU(Conv_{1\times1}(GAP(f))))) \tag{7}$$

where GAP represents the global average pooling. $Conv_{1\times1}$ indicates 1×1 convolutional layers. σ represents the sigmoid function. CA is the channel attention feature. Finally, we perform the addition operation on the spatial attention feature and the channel attention feature, then we can get the weight feature $f\prime$, which serves as the input attention map for the next GAM:

$$f\prime = Up(f \cdot (SA + CA)) \tag{8}$$

where Up represents the up-sample operation.

3.4 Context Boundary Restoration

As we all know, the context features contain rich semantic information, the fusion of context features is critical for restoring complete camouflaged objects. Therefore, we propose a context feature restoration module (CFRM) to aggregate the rich context features for improving the performance of COD.

Specifically, as shown in Fig.5, the CFRM consists of three branches (i.e., low-level features, global features and high-level features). First, we add a 3×3 convolutional layer to each of them. Then, the global features are used to guide the high-level features and the low-level features respectively through a concatenation operation. Next, global average pooling is used for the low-level features and the high-level features to get the pooled features f_{low}^{GAP} and f_{high}^{GAP}. Subsequently, the concatenation operation and a convolutional layer are adopted to gain the next global features f_g'.

$$f_{low}^{GAP} = GAP(Concat(Conv_{3\times3}(f_{low}), Up(Conv_{3\times3}(f_g)))) \tag{9}$$

$$f_{high}^{GAP} = GAP(Concat(Conv_{3\times3}(f_{high}), Conv_{3\times3}(f_g))) \tag{10}$$

$$f_g' = Conv_{3\times3}(Concat(f_{low}^{GAP}, f_{high}^{GAP})) \tag{11}$$

Simultaneously, in the low-level features and the high-level features, we leverage the element-wise addition operation to augment the missing context features. Then, another set of 3×3 convolutional layers are added to each of the two branches for obtaining the enhanced features f_{low}' and f_{high}'.

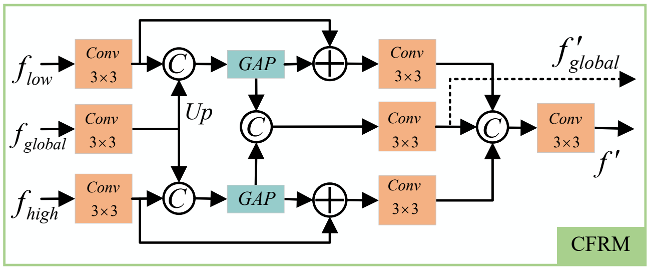

Figure 5: Detailed structure of the context feature restoration module (CFRM).

Finally, a concatenation operation and a 3×3 convolutional layer are applied to fuse these context features. This process can be formulated as:

$$f'_{low} = Conv_{3\times3}(f^{GAP}_{low} + Conv_{3\times3}(f_{low})) \tag{12}$$

$$f'_{high} = Conv_{3\times3}(f^{GAP}_{high} + Conv_{3\times3}(f_{high})) \tag{13}$$

$$f' = Conv_{3\times3}(Concat(f'_{low}, f'_{g}, f'_{high})) \tag{14}$$

In addition, high-level segmentation features contain rich semantic information and low-level texture features have a lot of boundary details. In order to obtain clear and whole object boundary, we introduce edge supervision. In particular, we fuse the high-level features with the low-level features and supervise them using edge maps. Specifically, we perform a residual channel attention block (RCAB) [Zhang *et al.*, 2018b] in the high-level features (i.e., f'_5) as the guiding information, and perform an element-wise multiplication operation with the low-level features (i.e., f'_2), followed by two 3×3 convolutional layers, a normalization layers and a ReLU layers to obtain the fused edge map f_{edge}. This process can be formulated as:

$$f_{edge} = CBR(CBR(RCAB(f'_5) + f'_2)) \tag{15}$$

where CBR denotes a 3×3 convolutional layer, BatchNorm2d and ReLU.

3.5 Loss Function

The binary cross entropy (BCE) loss and the intersection-over-union (IoU) loss are widely used in various image segmentation tasks. However, these losses treat each pixel equally and cannot distinguish the differences between pixels. In this paper, we use the weighted binary cross-entropy loss (wBCE) and IoU loss (wIoU), which can calculate the difference between the center pixel and its surrounding environment, and pay more attention on hard pixels to enhance the model generalization. In particular, we use the Consistency-Enhanced Loss (CEL) as an assistant, which consider the inter-pixel relationships and highlight the entire camouflaged region. To sum up, the loss function of our model is defined as follows:

$$L = L^{W}_{BCE} + L^{W}_{IoU} + \lambda L_{CEL} \tag{16}$$

where λ is a hyperparameter, it is set to 1 for balancing the contributions of the three losses.

In addition, we use the dice loss (L_{dice}) [Xie *et al.*, 2020] to address the strong imbalance between positive and negative samples. At last, the total loss can be formulated as:

$$L_{total} = L(P_{loc}, G) + L(P_{res}, G) + L_{dice}(P_{edge}, E) \tag{17}$$

where P_{loc} is the predicted location map, P_{res} is the predicted restoration map, G is the ground-truth map and E is the edge map.

4 Experiments

4.1 Datasets

We employ four widely-used COD benchmark datasets to evaluate our method, including: CHAMELEON [Skurowski *et al.*, 2018], CAMO [Le *et al.*, 2019], COD10K [Fan *et al.*, 2020a] and NC4K [Lv *et al.*, 2021]. Following the previous work [Fan *et al.*, 2021a], we use the combination of the train sets from CAMO and COD10K (4,040 images) as the training set, and evaluate on the rest ones.

4.2 Evaluation Metrics

Conventionally, we adopt four popular and standard metrics to evaluate the performance of our method: structure-measure (S_α) [Fan *et al.*, 2017], E-measure (E_ϕ) [Fan *et al.*, 2021b], weighted F-measure (F^{ω}_{β}) [Margolin *et al.*, 2014] and mean absolute error (M) [Perazzi *et al.*, 2012].

4.3 Implementation Details

We implement our model with PyTorch and adopt Res2Net-50 [Gao *et al.*, 2019] pre-trained on ImageNet as our backbone. We resize all the input images and ground-truths to 352×352 for both training and testing. During training, we set the batch size to 36 and use Adam algorithm to optimize the network parameters with a learning rate of $1e-4$, and decay it by 0.1 every 30 epochs. We apply random horizontal fliping, random cropping and random rotating to augment the training data. The training and testing processes are conducted on an NVIDIA Tesla V100 GPU (with 32GB memory) and Intel(R) Xeon(R) Gold 6,240, 2.60 GHz CPU device.

4.4 Comparison with State-of-the-Art Methods

We compare our PENet with 32 state-of-the-art COD methods, including CPD [Wu *et al.*, 2019a], PoolNet [Liu *et al.*, 2019], EGNet [Zhao *et al.*, 2019], SCRN [Wu *et al.*, 2019b], F^{3}Net [Wei *et al.*, 2020], CSNet [Gao *et al.*, 2020], SSAL [Zhang *et al.*, 2020b], UCNet [Zhang *et al.*, 2020a], MINet [Pang *et al.*, 2020], ITSD [Zhou *et al.*, 2020], PraNet [Fan *et al.*, 2020b], VST [Liu *et al.*, 2021], RCSB [Ke and Tsubono, 2022], SINet [Fan *et al.*, 2020a], R-MGL [Zhai *et al.*, 2021], TINet [Zhu *et al.*, 2021], UGTR [Yang *et al.*, 2021], PFNet [Mei *et al.*, 2021], SLSR [Lv *et al.*, 2021], UJSC [Li *et al.*, 2021], D^{2}C-Net [Wang *et al.*, 2021], SINet-V2 [Fan *et al.*, 2021a], C^{2}F-Net [Sun *et al.*, 2021], BgNet [Chen *et al.*, 2022b], SegMaR-1 [Jia *et al.*, 2022], BSA-Net [Zhu *et al.*, 2022], BGNet [Sun *et al.*, 2022], ER-RNet [Ji *et al.*, 2022], CubeNet [Zhuge *et al.*, 2022], Zoom-Net [Pang *et al.*, 2022], C^{2}F-Net-V2 [Chen *et al.*, 2022a] and FBNet [Lin *et al.*, 2023]. For a fair comparison, we obtain the

Mehtod	Year	CAMO-Test				CHAMELEON				COD10K-Test				NC4K			
		$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
Salient Object Detection / Medical Image Segmentation																	
CPD	2019	.716	.556	.113	.796	.857	.731	.048	.923	.750	.531	.053	.853	.787	.645	.072	.866
PoolNet	2019	.730	.575	.105	.819	.845	.690	.054	.933	.740	.506	.056	.844	.785	.635	.073	.865
EGNet	2019	.732	.604	.109	.820	.797	.649	.065	.065	.736	.517	.061	.854	.777	.639	.075	.864
SCRN	2019	.779	.643	.090	.850	.876	.741	.042	.939	.789	.575	.047	.880	.830	.698	.059	.897
F^3Net	2020	.711	.564	.109	.779	.848	.744	.047	.917	.739	.544	.051	.815	.780	.656	.070	.834
CSNet	2020	.771	.641	.092	.849	.856	.718	.047	.928	.778	.569	.047	.871	.750	.603	.088	.793
SSAL	2020	.644	.493	.126	.780	.757	.639	.071	.856	.668	.454	.066	.789	.699	.561	.093	.812
UCNet	2020	.739	.640	.094	.787	.880	.817	.036	.941	.776	.633	.042	.867	.811	.729	.055	.886
MINet	2020	.748	.637	.090	.838	.855	.771	.036	.937	.770	.608	.042	.859	.812	.720	.056	.887
ITSD	2020	.750	.610	.102	.830	.814	.662	.057	.901	.767	.557	.051	.861	.811	.679	.064	.883
PraNet	2020	.769	.663	.094	.837	.860	.763	.044	.935	.789	.629	.045	.879	.822	.724	.059	.888
VST	2021	.788	.734	.075	.840	.869	.786	.041	.891	.785	.654	.042	.836	.834	.772	.050	.878
RCSB	2022	.710	.581	.104	.737	.809	.701	.045	.860	.753	.590	.043	.802	.802	.705	.055	.841
Camouflaged Object Detection																	
SINet	2020	.751	.606	.100	.771	.869	.740	.044	.891	.771	.551	.051	.806	.808	.723	.058	.872
R-MGL	2021	.775	.673	.088	.847	.893	.813	.030	.923	.814	.666	.035	.865	.833	.739	.053	.893
TINet	2021	.781	.678	.087	.847	.874	.783	.038	.916	.793	.635	.043	.848	-	-	-	-
UGTR	2021	.784	.684	.086	.851	.888	.794	.031	.940	.817	.666	.036	.890	.839	.746	.052	.899
PFNet	2021	.782	.695	.085	.852	.882	.810	.033	.942	.800	.660	.036	.868	.829	.745	.053	.898
SLSR	2021	.787	.696	.080	.854	.890	.822	.030	.948	.804	.673	.037	.892	.840	.766	.048	.907
UJSC	2021	.800	.728	.073	.873	.891	.833	.030	.955	.809	.684	.035	.891	.842	.771	.047	.907
D^2C-Net	2021	.744	.735	.087	.818	.889	.848	.030	.939	.807	.720	.037	.876	-	-	-	-
SINet-V2	2021	.820	.743	.070	.882	.888	.816	.030	.942	.815	.680	.037	.887	-	-	-	-
C^2F-Net	2021	.796	.719	.080	.854	.888	.828	.032	.935	.813	.686	.036	.890	-	-	-	-
BgNet	2021	.804	.719	.075	.859	.885	.815	.032	.942	.804	.663	.039	.881	.843	.764	.048	.901
SegMaR-1	2022	.805	.724	.072	.864	.892	.823	.028	.937	.813	.682	.035	.880	-	-	-	-
BSA-Net	2022	.796	.717	.079	.851	.895	.841	.027	.946	.818	.699	.034	.891	-	-	-	-
BGNet	2022	.812	.749	.073	.870	-	-	-	-	.831	.722	.033	.901	.851	.788	.044	.907
ERRNet	2022	.761	.660	.088	.817	.877	.805	.036	.927	.780	.629	.044	.867	-	-	-	-
CubeNet	2022	.788	.682	.085	.838	.873	.787	.037	.928	.795	.644	.041	.864	-	-	-	-
ZoomNet	2022	.820	.752	.066	.877	**.902**	.845	**.023**	.943	**.838**	**.729**	**.029**	.888	.853	.784	.043	.896
C^2F-Net-V2	2022	.800	.730	.077	.869	.893	.845	.028	.947	.811	.691	.036	.891	-	-	-	-
FBNet	2023	.783	.702	.081	.839	.888	.828	.032	.939	.809	.684	.035	.889	-	-	-	-
PENet(Ours)	2023	**.828**	**.771**	**.063**	**.890**	**.902**	**.851**	.024	**.960**	.831	.723	.031	**.908**	**.855**	**.795**	**.042**	**.912**

Table 1: Quantitative comparison of our PENet and state-of-the-art methods for COD on four datasets, ↑ (or ↓) indicates that the higher (or the lower) the better. Best results are marked in **bold** fonts. SegMaR-1 is the 1-st iterative stage for fair comparison . "—": Not available.

results of these methods from the authors, ZoomNet [Pang *et al.*, 2022], or obtained by running the publicly available codes with well-trained models.

Quantitative Evaluation

Table 1 reports the detailed comparison results of PENet against other 32 state-of-the-art methods on four benchmark datasets. It can be seen that our proposed method consistently and significantly surpasses all the previous methods with a large margin on all four standard metrics. For example, compared with the simultaneously localize, segment and rank network SLSR [Lv *et al.*, 2021], our PENet improves the F_β^ω by 7.6%, 4.0%, 7.3% and 5.3% on the four dataset, respectively. It is worth mentioning that the results of our method is competitive with the C^2F-Net [Sun *et al.*, 2021] that uses the strategy of context-aware cross-level fusion, which boosts the S_α by 3.1%, 2.0% and 2.8% on the CAMO, CHAMELEON and COD10K dataset, respectively. In addition, our method surpasses UJSC [Li *et al.*, 2021] which even introduces extra SOD data for training.

Qualitative Evaluation

We provide several typical examples in Figure 6 , which visually shows the qualitative results of our PENet with other cutting-edge methods. It can be seen that most compared methods tend to detect some irrelevant surroundings or neglect some regions of camouflaged objects (e.g., the 2-nd and 3-rd rows). By contrast, the detection results of our PENet are more accurate and closest to the ground-truth annotations, including large camouflaged objects (e.g., the 5-th row), small camouflaged objects (e.g., the 4-th row) and low-contrast camouflaged objects (e.g., the 1-st row). These results visually demonstrate the superior performance of our method.

4.5 Ablation Study

In order to validate the effectiveness of each key module, we design a series of controlled experiments and present the results in Table 2.

Effectiveness of OLM. In Table 2, compared with the basic model (a), we can see that (b) outperforms (a) by 6.0%, 6.5%, 9.2% and 7.3% in terms of F_β^ω on the four dataset, respectively. This demonstrates that the OLM is effective for object localization and plays a key role in achieving high performance for COD tasks.

Effectiveness of GAM. From (b) and (c) in Table 2, we can see that our proposed GAM further improves the metric

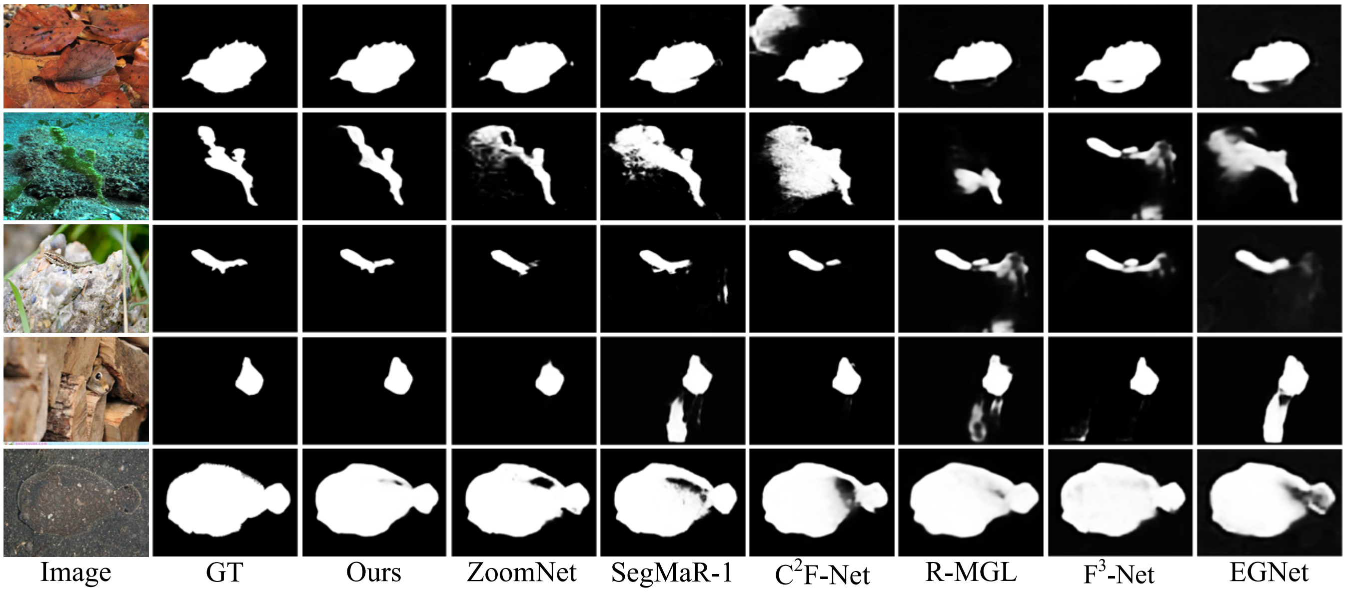

Figure 6: Qualitative comparisons of our PENet with state-of-the-art methods.

Ver.	Method	CAMO-Test				CHAMELEON				COD10K-Test				NC4K			
		$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$	$S_\alpha\uparrow$	$F_\beta^\omega\uparrow$	$M\downarrow$	$E_\phi\uparrow$
(a)	B	.790	.687	.083	.848	.862	.757	.040	.924	.780	.604	.045	.863	.818	.706	.057	.883
(b)	B+OLM	.818	.747	.070	.880	.888	.822	.028	.945	.821	.696	.034	.897	.850	.779	.045	.906
(c)	B+OLM+GAM	.820	.754	069	.879	.895	.838	.026	.954	.827	.714	.032	.902	.854	.791	.043	.911
(d)	B+OLM+GAM+CFRM w/o R	.825	.766	.066	.886	.898	.843	.025	.950	.830	.720	**.031**	.906	**.856**	**.795**	**.042**	.911
(e)	B+OLM+GAM+CFRM w/o E	822	.762	.067	.883	.896	.845	.025	.955	.829	.721	.031	.906	.855	.794	.042	.912
(f)	B+OLM+GAM+CFRM (Ours)	**.828**	**.771**	**.063**	**.890**	**.902**	**.851**	**.024**	**.960**	**.831**	**.723**	**.031**	**.908**	.855	**.795**	**.042**	**.912**

Table 2: Ablation analyses on the four datasets. Ver. = Version. “B” denotes removing all OLMs, GAMs and CFRMs, which just use the pre-trained models by a simple concatenation. “w/o R” means without RCAB. “w/o E” means without edge supervision.

results. Specifically, compared with (b) and (c), 0.7%, 1.6%, 1.8% and 1.2% performance improvement in terms of F_β^ω on the four dataset, respectively. This demonstrates that group attention module is beneficial for refining the multi-scale features and boosting the COD performance.

Effectiveness of CFRM. From Table 2, we observe that the results of model (d) outperforms (c). In particular, compared with the model (c), (d) increased the F_β^ω by 1.2%, 0.5%, 0.6% and 0.4% on the four dataset, respectively. This indicates that using CFRM to fuse contextual features can enhance the COD performance.

Effectiveness of RCAB. To verify the effectiveness of RCAB, we remove the RCAB in (d) and compare the results of (f) and (d). When RCAB is removed, the E_ϕ of (d) decreases by 0.4%, 1.0%, 0.2% and 0.1% on the four datasets, respectively. This implies that the RCAB module can leverage the global features of the image to guide the low-level features.

Effectiveness of edge supervision. To further validate the effectiveness of the edge supervision of our PENet, we compare the performance with and without the edge supervision. Specifically, compared with (f) and (e), we can find that with the edge supervision, the F_β^ω increases by 0.9%, 0.6%, 0.2% and 0.1% on the four dataset, respectively. This shows that edge supervision can guide the prediction to produce clear boundary.

5 Conclusion

In this paper, we are committed to addressing the challenges of accurate COD. We develop a “locate-refine-restore” strategy to gradually restore clear and complete camouflaged objects, which helps to improve the understanding and judgment of camouflaged objects. Specifically, we first propose an object location module (OLM) to initially locate the camouflaged region. Then, we design a group attention module (GAM) to enhance the texture feature representation. Finally, we introduce a context feature restoration module (CFRM) to restore the clear boundary by fusing the context features. We conduct extensive experiments on four benchmark datasets using four widely used evaluation metrics, which illustrates that our method can achieve state-of-the-art performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) [grant number 62101571], Natural Science Foundation of Hunan [grant number 2021JJ40685], and Hunan Provincial Innovation Foundation for Postgraduate [grant number QL20220018].

References

[Bazzani *et al.*, 2016] Loris Bazzani, Alessandra Bergamo, Dragomir Anguelov, and Lorenzo Torresani. Self-taught object localization with deep networks. In *WACV*, pages 1–9, 2016.

[Chen *et al.*, 2017] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2017.

[Chen *et al.*, 2020] Zuyao Chen, Qianqian Xu, Runmin Cong, and Qingming Huang. Global context-aware progressive aggregation network for salient object detection. In *AAAI*, volume 34, pages 10599–10606, 2020.

[Chen *et al.*, 2022a] Geng Chen, Si-Jie Liu, Yu-Jia Sun, Ge-Peng Ji, Ya-Feng Wu, and Tao Zhou. Camouflaged object detection via context-aware cross-level fusion. *IEEE TCSVT*, 32(10):6981–6993, 2022.

[Chen *et al.*, 2022b] Tianyou Chen, Jin Xiao, Xiaoguang Hu, Guofeng Zhang, and Shaojie Wang. Boundary-guided network for camouflaged object detection. *Knowledge-Based Systems*, 248:108901, 2022.

[Cuthill *et al.*, 2005] Innes C Cuthill, Martin Stevens, Jenna Sheppard, Tracey Maddocks, C Alejandro Párraga, and Tom S Troscianko. Disruptive coloration and background pattern matching. *Nature*, 434(7029):72–74, 2005.

[Dai *et al.*, 2021] Yimian Dai, Fabian Gieseke, Stefan Oehmcke, Yiquan Wu, and Kobus Barnard. Attentional feature fusion. In *WACV*, pages 3560–3569, 2021.

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

[Fan *et al.*, 2020a] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, pages 2777–2787, 2020.

[Fan *et al.*, 2020b] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation. In *MICCAI*, pages 263–273. Springer, 2020.

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

[Fan *et al.*, 2021b] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *SCIENTIA SINICA Informationis*, 6:6, 2021.

[Gao *et al.*, 2019] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, 43(2):652–662, 2019.

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

[Gokberk Cinbis *et al.*, 2014] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *CVPR*, pages 2409–2416, 2014.

[Hall *et al.*, 2013] Joanna R Hall, Innes C Cuthill, Roland Baddeley, Adam J Shohet, and Nicholas E Scott-Samuel. Camouflage, detection and identification of moving targets. *Proceedings of the Royal Society B: Biological Sciences*, 280(1758):20130064, 2013.

[Hu *et al.*, 2018] Xiaowei Hu, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, pages 7454–7462, 2018.

[Ji *et al.*, 2022] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu. Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognition*, 123:108414, 2022.

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

[Ke and Tsubono, 2022] Yun Yi Ke and Takahiro Tsubono. Recursive contour-saliency blending network for accurate salient object detection. In *WACV*, pages 2940–2950, 2022.

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

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

[Lin *et al.*, 2023] Jiaying Lin, Xin Tan, Ke Xu, Lizhuang Ma, and Rynson WH Lau. Frequency-aware camouflaged object detection. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(2):1–16, 2023.

[Liu *et al.*, 2018] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *ECCV*, pages 385–400, 2018.

[Liu *et al.*, 2019] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, pages 3917–3926, 2019.

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

[Lv *et al.*, 2021] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, pages 11591–11601, 2021.

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

[Mei *et al.*, 2020] Haiyang Mei, Xin Yang, Yang Wang, Yuanyuan Liu, Shengfeng He, Qiang Zhang, Xiaopeng Wei, and Rynson WH Lau. Don't hit me! glass detection in real-world scenes. In *CVPR*, pages 3687–3696, 2020.

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

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

[Pang *et al.*, 2022] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *CVPR*, pages 2160–2170, 2022.

[Perazzi *et al.*, 2012] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.

[Price *et al.*, 2019] Natasha Price, Samuel Green, Jolyon Troscianko, Tom Tregenza, and Martin Stevens. Background matching and disruptive coloration as habitat-specific strategies for camouflage. *Scientific reports*, 9(1):1–10, 2019.

[Skurowski *et al.*, 2018] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018.

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

[Sun *et al.*, 2022] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection. In *IJCAI*, pages 1335–1341, 2022.

[Wang *et al.*, 2021] Kang Wang, Hongbo Bi, Yi Zhang, Cong Zhang, Ziqi Liu, and Shuang Zheng. d^2c-net: A dual-branch, dual-guidance and cross-refine network for camouflaged object detection. *IEEE Transactions on Industrial Electronics*, 69(5):5364–5374, 2021.

[Wei *et al.*, 2020] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: fusion, feedback and focus for salient object detection. In *AAAI*, volume 34, pages 12321–12328, 2020.

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

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

[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*, pages 696–711. Springer, 2020.

[Yang *et al.*, 2021] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *ICCV*, pages 4146–4155, 2021.

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

[Zhang *et al.*, 2018a] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, pages 1325–1334, 2018.

[Zhang *et al.*, 2018b] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018.

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

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

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

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

[Zhu *et al.*, 2021] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu. Inferring camouflaged objects by texture-aware interactive guidance network. In *AAAI*, volume 35, pages 3599–3607, 2021.

[Zhu *et al.*, 2022] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jing Qin. I can find you! boundary-guided separated attention network for camouflaged object detection. In *AAAI*, volume 36, pages 3608–3616, 2022.

[Zhuge *et al.*, 2022] Mingchen Zhuge, Xiankai Lu, Yiyou Guo, Zhihua Cai, and Shuhan Chen. Cubenet: X-shape connection for camouflaged object detection. *Pattern Recognition*, 127:108644, 2022.