

# Camouflaged Object Detection That Does Not Require Additional Priors

Yuchen Dong <sup>1</sup>, Heng Zhou <sup>1</sup>, Chengyang Li <sup>1,2</sup>, Junjie Xie <sup>1</sup>, Yongqiang Xie <sup>1,\*</sup> and Zhongbo Li <sup>1,\*</sup>

<sup>1</sup> Institute of Systems Engineering, Academy of Military Sciences, Beijing 100071, China; dongyc91@outlook.com (Y.D.); zh.ams@foxmail.com (H.Z.); chengyang\_li@stu.pku.edu.cn (C.L.); xiejunjie06@163.com (J.X.)

<sup>2</sup> School of Computer Science, Peking University, Beijing 100871, China

\* Correspondence: yqxie2021@outlook.com (Y.X.); lzb05296@163.com (Z.L.)

**Abstract:** Camouflaged object detection (COD) is an arduous challenge due to the striking resemblance of camouflaged objects to their surroundings. The abundance of similar background information can significantly impede the efficiency of camouflaged object detection algorithms. Prior research in this domain has often relied on supplementary prior knowledge to guide model training. However, acquiring such prior knowledge is resource-intensive. Furthermore, the additional provided prior information is typically already embedded in the original image, but this information is underutilized. To address these issues, in this paper, we introduce a novel Camouflage Cues Guidance Network (CCGNet) for camouflaged object detection that does not rely on additional prior knowledge. Specifically, we use an adaptive approach to track the learning state of the model with respect to the camouflaged object and dynamically extract the cues of the camouflaged object from the original image. In addition, we introduce a foreground separation module and an edge refinement module to effectively utilize these camouflage cues, assisting the model in fully separating camouflaged objects and enabling precise edge prediction. Extensive experimental results demonstrate that our proposed methods can achieve superior performance compared with state-of-the-art approaches.

**Keywords:** camouflaged object detection (COD); deep learning; without prior knowledge; camouflage cues



**Citation:** Dong, Y.; Zhou, H.; Li, C.; Xie, J.; Xie, Y.; Li, Z. Camouflaged Object Detection That Does Not Require Additional Priors. *Appl. Sci.* **2024**, *14*, 2621. <https://doi.org/10.3390/app14062621>

Academic Editor: Sungho Kim

Received: 10 February 2024

Revised: 9 March 2024

Accepted: 11 March 2024

Published: 21 March 2024



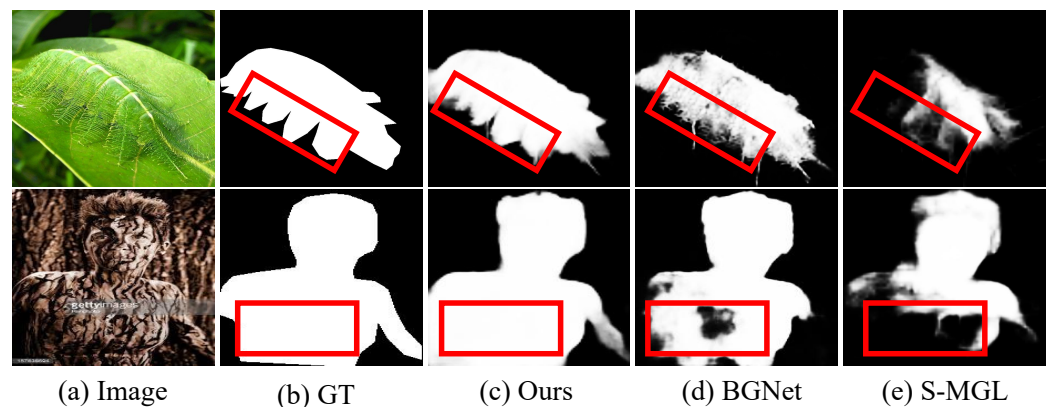
**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Deep learning has made significant strides in the field of object detection [1–3]. In object detection research, scholars primarily emphasize detecting objects in ordinary scenes [4]. However, as deep learning algorithms have become increasingly industrialized in recent years, the research community has started to emphasize the practicality of camouflaged object detection (COD). Specifically, COD is a research field that focuses on identifying camouflaged objects within images [5,6]. Camouflaged objects can be categorized into two main types: natural and artificial camouflage [7]. Natural camouflage refers to how organisms utilize brightness, color, and body shape to conceal themselves in their environment [8]. Many species employ this strategy to survive, either to avoid predators or to capture prey [9,10]. On the other hand, artificial camouflage involves using techniques such as paint, clothing, and accessories to create visual deception and conceal the object. This form of camouflage is commonly employed in military applications [11,12], art [13], and other domains [14,15]. The primary objective of camouflaged object detection is to generate a binary detection map for the camouflaged objects in the image [7]. However, detecting camouflage presents a challenge due to the visual similarity between the foreground and background [16–18].

Early camouflaged object detection algorithms drew inspiration from natural processes such as animal predation and used a two-part model of search and detection [19]. These early algorithms relied solely on basic images for training. Subsequent researchers concluded that training on image features alone was insufficient to identify and understand

camouflaged objects accurately [20]. Therefore, researchers introduced additional prior knowledge during model training. Numerous COD algorithms [20–22] have achieved commendable results by relying on the guidance of prior knowledge. While prior knowledge has made significant contributions, the drawback lies in the manual labeling requirement, incurring substantial costs. Meanwhile, we have discovered that satisfactory results can be achieved without relying on any prior knowledge, using only cues extracted from the original image to guide model training. As depicted in Figure 1, our prediction results outperform BGNet [20] and S-MGL [21], which exhibit unsatisfactory performance in predicting the boundaries of the object even with the incorporation of boundary priors during training. Furthermore, owing to the inadequate understanding of the camouflaged object’s overall structure, there exists a significant issue of uneven pixel distribution in the prediction results.



**Figure 1.** Visual examples of camouflaged object detection in challenging scenarios showcase our algorithm’s capabilities. When compared to recent COD algorithms (e.g., BGNet, and S-MGL) that rely on boundary priors, our algorithm achieves superior boundary detection results without the need for any prior information and effectively mitigates uneven pixel distribution.

Based on the above issues, we propose a Camouflage Cues Guidance Network (CCGNet) for camouflaged object detection. Specifically, we introduce the adaptive feature fusion Module (AFFM), which vigilantly monitors the model’s learning state and dynamically selects and integrates features to generate cues for camouflage. The camouflage cues comprise the available knowledge from each model layer, which can be employed to rectify and complement features across different layers and direct the model’s attention toward the overall image structure. In order to effectively utilize the camouflage cues, we introduce the foreground separation module (FSM). This module leverages the camouflage cues to complement the multi-layer features, allowing the model to focus more on the overall structure of the camouflaged objects and effectively separate them from similar environments. Additionally, to refine the model’s representation further, especially the edge information, we propose the edge refinement module (ERM). ERM achieves finer edge predictions by combining features with contextual information. As shown in Figure 1, in comparison to other COD algorithms, we achieve visually superior edge delineation results and mitigate uneven pixel distribution without the use of priors.

Overall, our work makes the following main contributions:

- For the camouflaged object detection (COD) problem, we introduce a novel Camouflage Cues Guidance Network, i.e., CCGNet. This network incorporates an Adaptive Feature Fusion Module (AFFM) to enrich the model’s comprehension of the overall structure of camouflaged objects by effectively extracting and integrating the inherent semantic information present within the image itself.
- We introduce two crucial modules, namely, the foreground separation module (FSM) and the edge refinement module (ERM). These modules utilize the camouflage cues

generated by the AFFM to thoroughly investigate the relevant semantic details within the image and improve the edge representation of camouflaged objects.

- Extensive experiments conducted on three traditional benchmark datasets show that our model outperforms state-of-the-art models in all four metrics.

## 2. Related Work

### 2.1. Camouflaged Object Detection Dataset

Several datasets have been developed to facilitate advancements in camouflaged object detection. Commonly used datasets include CAMO [23], COD10K [19], and NC4K [22], which comprise 1250, 10,000, and 4121 images, respectively. Additionally, Skurowski et al. [24] introduced the CHAMELEON dataset, consisting of 76 images. Lv et al. [18,22] presented the CAM-FR dataset, which includes 2280 images, each annotated with localization and ranking. Zheng et al. [11] introduced the first camouflaged human dataset, comprising 1000 images. Subsequently, Fang et al. [12] expanded upon Zheng et al.'s dataset, resulting in a camouflaged human dataset with 2600 images. In this article, we employed the three most commonly used datasets, namely CAMO, COD10K, and NC4K.

### 2.2. Camouflaged Object Detection

In recent years, COD methods can be categorized into two groups: those utilizing prior information and those not relying on prior information.

**With additional prior information.** Lv et al. [18,22] introduced a joint framework for locating, segmenting, and ranking camouflaged objects. They incorporated additional ranking information during training, improving camouflage understanding. Zhai et al. [21] proposed mutual graph learning, which effectively separates images into task-specific feature maps for precise localization and boundary refinement. Sun et al. [20] explored the use of object-related edge semantics as an additional guide for model learning, encouraging the generation of features that emphasize the object's edges. He et al. [25] suggested using edge likelihood maps for guiding the fusion of camouflaged object features, aiming to enhance detection performance by improving boundary details. Kajiura et al. [26] employed a pseudo-edge generator to predict edge labels, contributing to accurate edge predictions. Zhu et al. [27] proposed the utilization of Canny edge [28] and Conedge techniques to assist in model training. Li et al. [29] proposed co-training camouflage and saliency objects [30,31] to enhance model detection. Yang et al. [5] integrated the advantages of Bayesian learning and transformer-based [32,33] inference. They introduced uncertainty-guided random masking as prior knowledge to facilitate model training. Bian et al. [34] utilized edge information to assess the degree of concealment of the object. Song et al. [35] suggested the selection of certain structural features, such as illumination, texture direction, and edges, and employed weighted structural texture similarity to assess the impact of camouflage texture. However, prior information is often expensive and impractical.

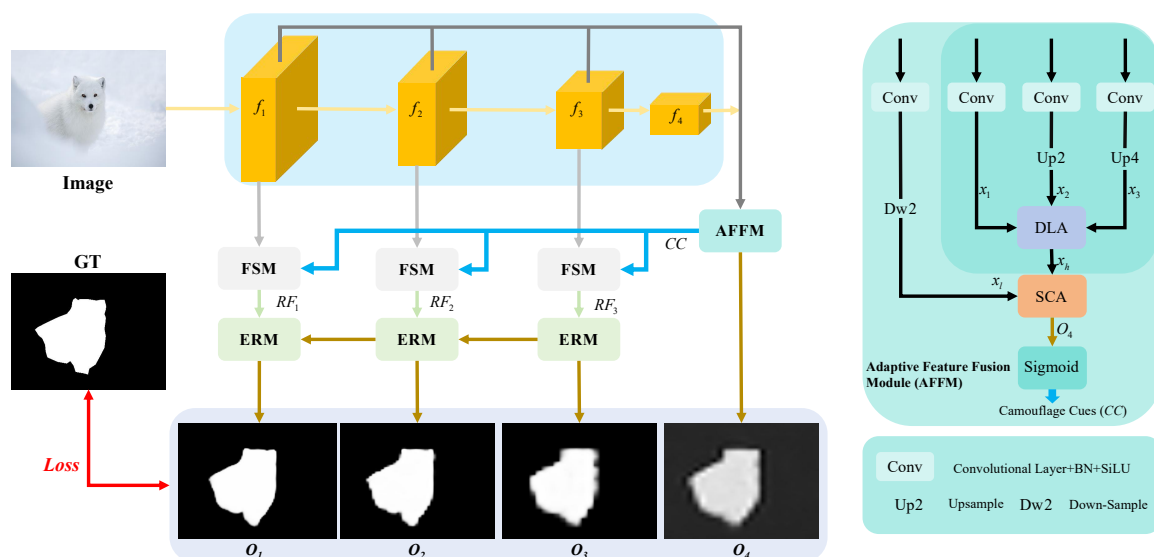
**Without additional prior information.** Mei et al. [36] proposed a localization module, a focus module, and a novel distraction mining strategy to enhance model performance. Fan et al. [7,19] introduced a search and recognition network inspired by the predatory behavior of hunters in nature, which implements object localization and recognition steps. Sun et al. [37] proposed an attention-inducing fusion module that integrates multilevel features and incorporates contextual information for more effective prediction. Zhang et al. [10] proposed a model that incorporates two processes of predation, specifically sensory and cognitive mechanisms. To achieve this, specialized modules were designed to selectively and attentively aggregate initial features using an attention-based approach. Jia et al. [38] proposed a method where the model attends to fixation and edge regions and utilizes an attention-based sampler to progressively zoom in on the object region instead of increasing the image size. This approach allows for the iterative refinement of features. Ren et al. [39] introduced the concept of constructing multiple texture-aware refinement modules within a deep convolutional neural network. These modules aim to learn texture-aware features that can accentuate subtle texture differences between the

camouflaged object and the background. Dong et al. [40] integrated large receptive fields and effective feature fusion into a unified framework to enhance the model's ability to detect camouflaged objects. In practice, algorithms that do not depend on prior information typically utilize various techniques to aggregate features with different receptive field sizes to obtain better detection results. Many of these algorithms exhibit inefficiency and frequently encounter limitations in the efficient extraction of image information. In contrast, our proposed CCGNet effectively guides the model training process by autonomously extracting valuable features from the image itself, generating camouflage cues without the need for external prior knowledge.

### 3. Proposed Method

#### 3.1. Overall Architecture

The architecture of CCGNet is illustrated in Figure 2, which consists of three modules: the adaptive feature fusion module (AFFM), foreground separation module (FSM), and edge refinement module (ERM), as described in Section 3.2, Section 3.3, and Section 3.4 respectively. For extracting multiscale features, the Res2Net-50 [41] architecture is utilized as the backbone. In this paper, the multiscale features are obtained from the last four layers of the feature hierarchy. The layer closest to the input is excluded as it contains excessive noise and has a small receptive field. Please note that the layer closest to the input is not depicted in Figure 2.



**Figure 2. Overview of our framework.** The proposed CCGNet framework comprises three components: the adaptive feature fusion module (AFFM), the foreground separation module (FSM), and the edge refinement module (ERM). The AFFM assumes a pivotal role in delving into the overall structure of camouflaged objects through the adaptive fusion of multilayer features, thus generating tailored camouflage cues (CC) that align with the model's learning state. The combined operation of the foreground separation module (FSM) and the edge refinement module (ERM), in tandem with the camouflage cues, results in a significant enhancement of feature representation.

#### 3.2. Adaptive Feature Fusion Module

Boundary priors and ranking priors have been employed to assist object detection models [18,22,42]. Nevertheless, the fundamental information utilized for detection resides within image features. Given the inherent resemblance between foreground and background features in camouflage images and the potential loss of crucial information during model training, effectively harnessing reliable feature information has proven to be a significant challenge for previous models. In such cases, the integration of additional prior knowledge can indeed lead to substantial improvements in detection performance.

However, these supplementary prior features rely on human identification, resulting in increased labor costs, and excessive dependence on this supplementary prior information can potentially hinder the algorithm's adaptability and effectiveness. It is crucial to emphasize that the training data already contain supplementary prior knowledge, which may not be fully retained or optimally utilized due to the model's design. To obtain comprehensive and beneficial information for camouflaged object detection, we introduce the adaptive feature fusion module (AFFM). This module dynamically fuses multilayer features based on the model's learning state regarding the camouflaged object, extracting valuable knowledge for detection. This process ultimately yields comprehensive camouflage cues (CC). The camouflage cues encompass all the knowledge that the model has learned, which is beneficial for camouflaged object detection. The efficient utilization of these camouflage cues can enhance the model's understanding of the overall structure of the camouflaged object. The details of AFFM are described below.

As shown in Figure 2, a convolution operation is applied to all input features. Subsequently, all input features are resized. The high-level feature  $\{f_i\}_{i=2}^4$  is adjusted to size  $\{x_i\}_{i=1}^3 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 256}$ , and the low-level feature  $f_1$  is adjusted to size  $x_l \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 128}$ . Following that, we leverage the deep layer attention (DLA) [43] mechanism to augment the model's comprehension of the overall structure of camouflaged objects. This involves analyzing the interplay between each feature layer, assigning weights to individual layers according to the significance of the acquired features, and applying weight filtering to extract features pertinent to the camouflaged object. Moreover, within the context of DLA, weight generation entails the computation of correlations between feature layers. These feature layers adeptly encapsulate the current model's learning state with respect to the camouflaged object. Consequently, our feature fusion process closely aligns with the model's learning state concerning the camouflaged object. The computation process of DLA is represented by Equation (1).

$$\begin{aligned} w_{i,j} &= \text{Softmax}(\phi(x)_i \cdot (\phi(x))_j^T), i, j \in \{1, 2, 3\} \\ x_j &= \beta \sum_{i=1}^3 w_{i,j} x_i + x_j, x_i / x_j \in \{x_1, x_2, x_3\} \\ x_h &= [x_1; x_2; x_3] \end{aligned} \quad (1)$$

where  $w_{i,j}$  represents the correlation weight between layer  $i$  and layer  $j$ ,  $\phi(\cdot)$  denotes the reshape operations, and  $\beta$  is initially set to 0 and then automatically assigned by the network.

Then, we fuse the obtained feature  $x_h \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 768}$  and feature  $x_l$ . Inspired by [20], we employ the spatial channel attention (SCA) mechanism to investigate the correlations among different feature channels and extract valuable detection knowledge from them. The above process can be described as follows:

$$\begin{aligned} O_4 &= \text{SCA}(x_l, x_h) \\ \text{SCA} &\Leftarrow \text{Conv}_{1 \times 1}(\text{Conv}_{3 \times 3}(\sigma(\text{Conv}_{1 \times 1}(\cdot)))) \end{aligned} \quad (2)$$

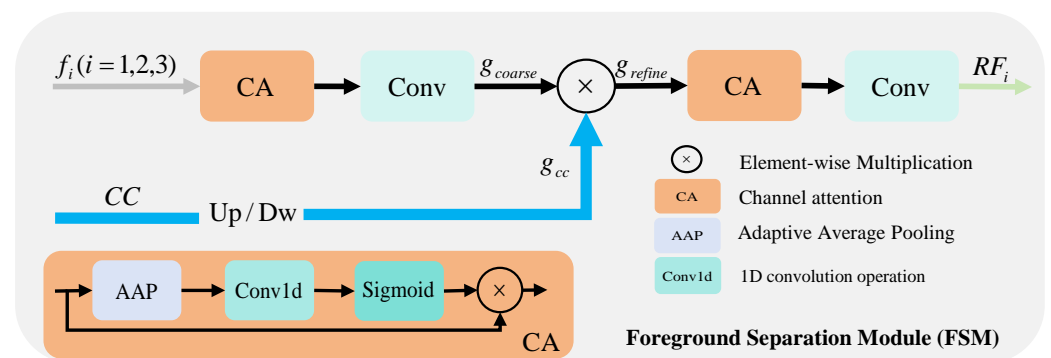
where  $\text{Conv}_{i \times i}$  represents a set of convolution operations with a convolution kernel size of  $i \times i$ , a BN (batch normalization) layer, and a SiLU activation function.  $\sigma$  denotes CBAM [44].

Finally, we apply the sigmoid function to  $O_4$  to obtain the camouflage cues (CC), which is represented as a binary graph. The camouflage cues that fuse high-level and low-level feature generation in AFFM contain a wealth of knowledge about the model's comprehension of the overall structure of the camouflaged object that can be used for camouflaged object detection. By combining features from different layers, the model takes advantage of the different advantages inherent in each layer. This adaptive fusion process ensures that the generated camouflage cues are consistent with the current stage of the

model’s understanding of the camouflaged object, providing valuable guidance throughout the model’s training process.

### 3.3. Foreground Separation Module

As mentioned earlier, AFFM extracts cues related to camouflaged objects from images to construct CC. This information serves as a guide for model training and supplements the missing knowledge in each layer. In order to effectively utilize CC, we designed the foreground separation module (FSM). The main goal of this module is to achieve a complete prediction of camouflaged objects by incorporating learning cues specific to camouflaged objects into the representation learning process and effectively separating the camouflaged object from the background in the image. More precisely, using CC to enhance the features of each layer can enhance the model’s learning of the overall structure of the camouflaged object and effectively alleviate the problem of uneven pixel distribution. The overall structure of FSM is depicted in Figure 3.



**Figure 3.** Illustration of FSM. The foreground separation module (FSM) fully segregates the foreground information of the camouflaged object through multifeature channel filtering, utilizing CC to complete the missing information in each feature layer.

Specifically, inspired by [45], we use channel attention (CA) for image features  $\{f_i\}_{i=1}^3$  to explore key channel features, and then obtain  $g_{coarse} \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times 256}$  through a convolution operation. The  $g_{refine}$  is derived from  $g_{coarse}$  and  $g_{cc}$ .  $g_{cc}$  is obtained by applying an up- or downsampling operation to CC. Certainly,  $g_{refine}$  already encompasses fairly comprehensive information about the object. However, to attain precise separation of foreground and background features, we re-examine the channel features to filter out any camouflage features. Finally, the final output  $\{RF_i\}_{i=1}^3$  is obtained by using  $1 \times 1$  convolution.

The calculation process is depicted in Equation (3).

$$\begin{aligned}
 g_{coarse} &= Conv_{3 \times 3}(CA(f_i)), i \in \{1, 2, 3\} \\
 g_{refine} &= g_{coarse} \otimes Up_2/Dw_2(g_{cc}) \\
 RF_i &= Conv_{1 \times 1}(CA(g_{refine})), i \in \{1, 2, 3\}
 \end{aligned}
 \tag{3}$$

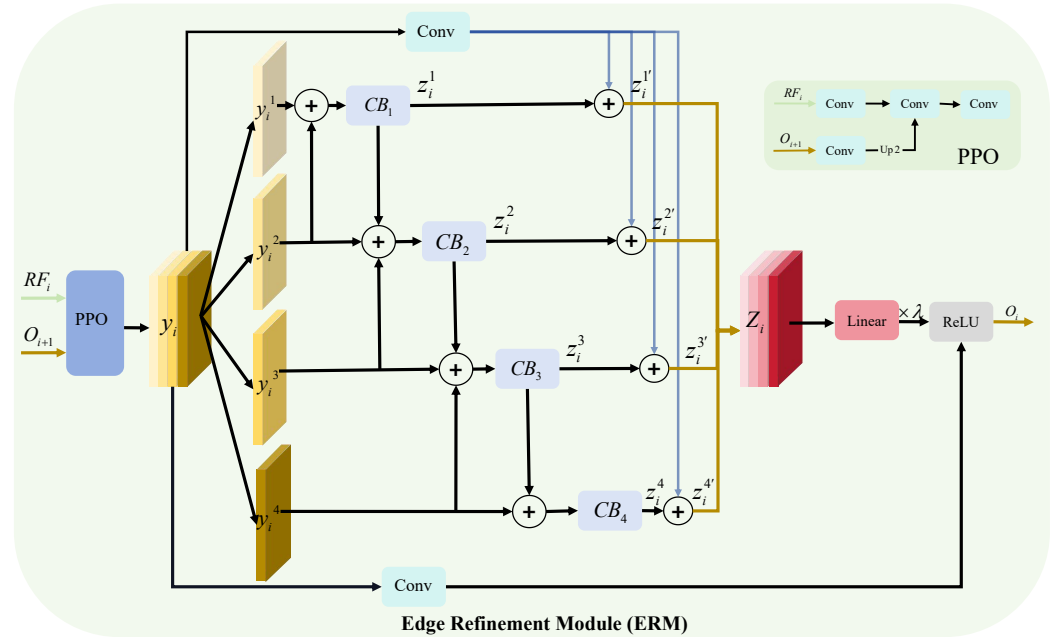
where  $f_i$  represents the image features output by the backbone.  $RF_i$  is the refined feature.

### 3.4. Edge Refinement Module

While FSM can effectively utilize the camouflage cues to complement and refine the features in each layer, the perception of specific details, such as edge information, is still not precise enough. To address this issue without relying on prior knowledge to supervise model training, we introduce the edge refinement module (ERM), which filters features to help the model achieve finer edge predictions by exploring contextual information. In contrast to the texture enhancement module (TEM), our approach also considers the semantic correlations between different branches within the same feature layer.

As shown in Figure 4, we obtain features  $\{y_i\}_{i=1}^3 \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times 256}$  by aggregating the fine features  $\{RF_i\}_{i=1}^3$  and the high-level output features  $\{O_i\}_{i=2}^3$  through the preprocessing operation (PPO). The feature  $\{y_i\}_{i=1}^3$  comprises top-down semantic information. The generation process of  $\{y_i\}_{i=1}^3$  is illustrated in Equation (4).

$$\begin{aligned} y_i &= PPO(RF_i, O_{i+1}), i \in \{1, 2\} \\ y_3 &= RF_3 \end{aligned} \tag{4}$$



**Figure 4.** Illustration of ERM. ERM learns the interrelation between various channels of the same feature and connects contextual features to achieve refined boundary predictions.

To explore the semantic correlation between different channel branches of the same layer feature, we divide  $\{y_i\}_{i=1}^3$  into four parts  $[y_i^1; y_i^2; y_i^3; y_i^4]$  in the channel dimension. Inspired by [20], we add the features of a branch to the features of its neighboring branches. This process can be formulated as follows:

$$\begin{aligned} z_i^1 &= CB_1(y_i^1 + y_i^2), i \in \{1, 2, 3\} \\ z_i^j &= CB_j(z_i^{j-1} + y_i^j + y_i^{j+1}), i \in \{1, 2, 3\}, j \in \{2, 3\} \\ z_i^4 &= CB_4(z_i^3 + y_i^4), i \in \{1, 2, 3\} \end{aligned} \tag{5}$$

where  $\{CB_j\}_{j=1}^4$  indicates a series of convolution operations and the specific composition of  $\{CB_j\}_{j=1}^4$  is shown in Equation (6).

$$\begin{aligned} CB_1 &\Leftarrow DConv_{3 \times 3}^1(Conv_{1 \times 1}(\cdot)) \\ CB_2 &\Leftarrow DConv_{3 \times 3}^3(Conv_{3 \times 1}(Conv_{1 \times 1}(\cdot))) \\ CB_3 &\Leftarrow DConv_{3 \times 3}^3(Conv_{1 \times 3}(Conv_{1 \times 1}(\cdot))) \\ CB_4 &\Leftarrow DConv_{3 \times 3}^5(Conv_{1 \times 3}(Conv_{3 \times 1}(Conv_{1 \times 1}(\cdot)))) \end{aligned} \tag{6}$$

where  $DConv_{i \times i}^j$  represents a atrous convolution [46] with a convolution kernel size of  $i \times i$  and a dilation rate of  $j$ .

In order to avoid losing important detection information during convolution, we add a residual structure to each interaction branch to obtain feature  $z_i^j \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times 256}$ . Merging

all  $z_i^j$  together, we obtain feature  $\{Z_i\}_{i=1}^3 \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times 256}$ , which incorporates semantic information from adjacent branches. It filters features by learning the relationships between adjacent features, thereby prompting the model to focus more on expressing details. As illustrated in Figure 2, as the feature progresses forward,  $O_1$  displays finer edge details compared to  $\{O_i\}_{i=2}^4$ .

Then, we perform a series of computations for  $\{Z_i\}_{i=1}^3$  to obtain the output feature  $\{O_i\}_{i=1}^3 \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times 1}$ . The computation process is as follows:

$$O_i = y_i + \lambda \cdot R(L(Z_i)), i \in \{1, 2, 3\}. \quad (7)$$

where  $R$  denotes the ReLU function,  $L$  denotes the linear function, and  $\lambda$  is the scaling factor.

### 3.5. Loss Function

CCGNet incorporates two types of loss functions: dice loss ( $L_{dice}$ ) [14] and structural loss ( $L_{struct}$ ) [47]. For  $O_4$ , we utilize  $L_{dice}$  to balance scenarios where positive and negative samples are unbalanced. For  $\{O_i\}_{i=1}^3$ , we apply  $L_{struct}$  to promote structural consistency and accuracy.

$$L_{struct} = L_{BCE}^w + L_{IoU}^w \quad (8)$$

Therefore, the total loss is defined as in Equation (9).

$$L_{total} = \sum_{i=1}^3 L_{struct}(O_i, GT) + L_{dice}(O_4, GT) \quad (9)$$

where  $\{O_i\}_{i=1}^4$  represents the feature map generated by CCGNet, and GT refers to the ground truth. During the testing process,  $O_1$  is used as the prediction result of the model.

## 4. Experiments and Analysis

### 4.1. Experiment Setup

We implement our model using PyTorch and utilize the Adam optimization algorithm [48] to optimize the overall parameters. The learning rate starts at  $1 \times 10^{-4}$ , dividing by 10 every 50 epochs. The model is accelerated using an NVIDIA 3090Ti GPU. During the training stage, the batch size is set to 36, and the whole training takes approximately 100 epochs. And the scaling factor  $\lambda$  is set to 0.5.

### 4.2. Comparison with States of the Art

We evaluated our method on three benchmark datasets: CAMO [23], COD10K [19], and NC4K [22]. To gauge the effectiveness of our approach, we employed four widely recognized and standardized metrics: MAE ( $\mathcal{M}$ ) [31], weighted F-measure ( $F_\beta^w$ ) [49], average E-measure ( $E_\phi$ ) [50], and S-measure ( $S_\alpha$ ) [51].

To demonstrate the effectiveness of our proposed CCGNet, we conducted a comparative analysis by comparing its prediction results with those of eleven state-of-the-art methods. The selected methods for comparison are EGNet [42], SCRNet [52], F<sup>3</sup>Net [47], CSNet [53], BASNet [54], SINet [19], PFNet [36], S-MGL [21], BGNet [20], LSR+ [18] and C<sup>2</sup>FNet [37]. For a fair comparison, the prediction results of these methods were provided by the original authors or generated using models trained with open-source code.

**Quantitative Evaluation.** Table 1 summarizes the quantitative results of different COD methods on the three benchmark datasets. Our method outperforms previous methods in all four evaluation indicators on the three datasets. In particular, compared with the BGNet [20], our method shows an average increase of 1.13% in  $S_\alpha$ , 0.16% in  $E_\phi$ , and 0.1% in  $F_\beta^w$ .

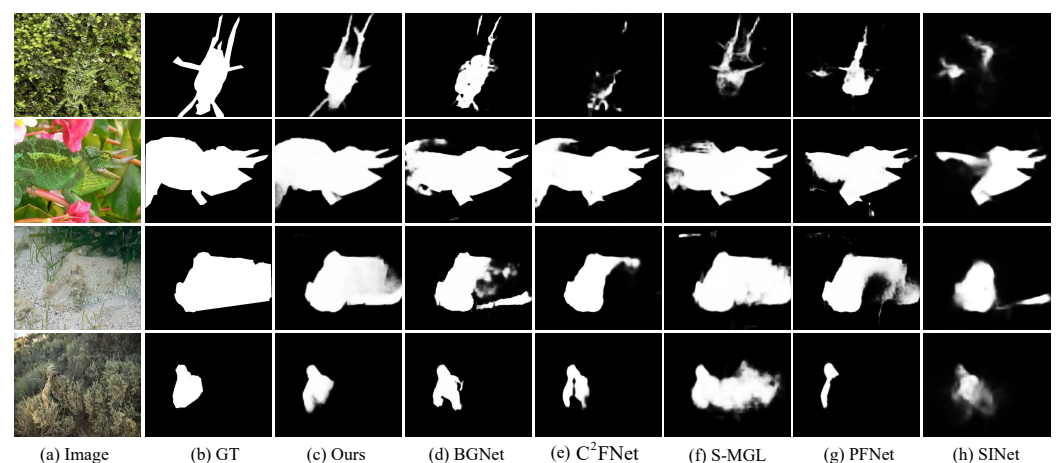
Regarding the slightly lower training results observed in comparison to BGNet for certain datasets, we conducted a comprehensive analysis to identify the underlying reasons for these disparities. As illustrated in Figure 5, when we compare the prediction results of CCGNet with those of other models, we observe that CCGNet effectively mitigates the issue of uneven pixel distribution. However, this occasionally leads to a broader range of

predicted objects, which may have a slight influence on specific evaluation metrics. Notable examples include the legs of insects in the first row of Figure 5 and the size of people in the fourth row.

**Table 1.** Quantitative comparison with state-of-the-art methods for COD on three benchmarks using four widely used evaluation metrics (i.e.,  $S_\alpha$ ,  $E_\phi$ ,  $F_\beta^w$ , and  $\mathcal{M}$ ). The best scores are highlighted in **bold**, and the symbol  $\uparrow$  indicates that a higher score is better.

Method	CAMO-Test				COD10K-Test				NC4K			
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
2019 EGNNet [42]	0.732	0.800	0.604	0.109	0.736	0.810	0.517	0.061	0.777	0.841	0.639	0.075
2019 SCRNet [52]	0.779	0.797	0.643	0.090	0.789	0.817	0.575	0.047	0.830	0.854	0.698	0.059
2020 F <sup>3</sup> Net [47]	0.711	0.741	0.564	0.109	0.739	0.795	0.544	0.051	0.780	0.824	0.656	0.070
2020 CSNet [53]	0.771	0.795	0.642	0.092	0.778	0.809	0.569	0.047	0.750	0.773	0.603	0.088
2020 BASNet [54]	0.749	0.796	0.646	0.096	0.802	0.855	0.677	0.038	0.817	0.859	0.732	0.058
2020 SINet [19]	0.745	0.804	0.644	0.092	0.776	0.864	0.631	0.043	0.808	0.871	0.723	0.058
2021 PFNet [36]	0.782	0.841	0.695	0.085	0.800	0.877	0.660	0.040	0.829	0.887	0.745	0.053
2021 S-MGL [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
2021 C <sup>2</sup> FNet [37]	0.799	0.851	0.710	0.078	0.811	0.886	0.669	0.038	0.843	0.899	0.757	0.050
2023 LSR+ [18]	0.789	0.840	0.751	0.079	0.805	0.880	0.711	0.037	0.840	0.896	<b>0.801</b>	0.048
2022 BGNNet [20]	0.807	0.861	0.742	0.072	0.829	<b>0.898</b>	<b>0.719</b>	0.033	0.849	<b>0.903</b>	0.785	0.045
CCGNet (Ours)	<b>0.827</b>	<b>0.877</b>	<b>0.754</b>	<b>0.069</b>	<b>0.833</b>	0.888	0.710	<b>0.033</b>	<b>0.859</b>	0.902	0.785	<b>0.045</b>

**Qualitative Evaluation.** We qualitatively evaluated different COD algorithms on the CAMO dataset. As shown in Figure 5, our model provides more accurate details for the camouflaged object in the second row, which can be attributed to the precise localization and comprehensive appearance prediction. This result indicates that our FSM and ERM help the model separate the foreground background better and refine the edge prediction. For the fish in the third row, the current COD algorithm achieves accurate localization but predicts the overall structure poorly, resulting in uneven pixel distributions. Specifically, the phenomenon of missing object structures and labeling the foreground as the background occurs in the predictions. In contrast, our predictions show significantly less uneven pixel distribution, which suggests that our proposed AFFM is helpful for the model in understanding the overall structure of the object. For the fourth row of people, our model achieves more accurate localization and boundary prediction compared to other COD algorithms. These results show that our proposed CCGNet can understand the object's structure better and achieve finer edge prediction than COD algorithms using additional priors.

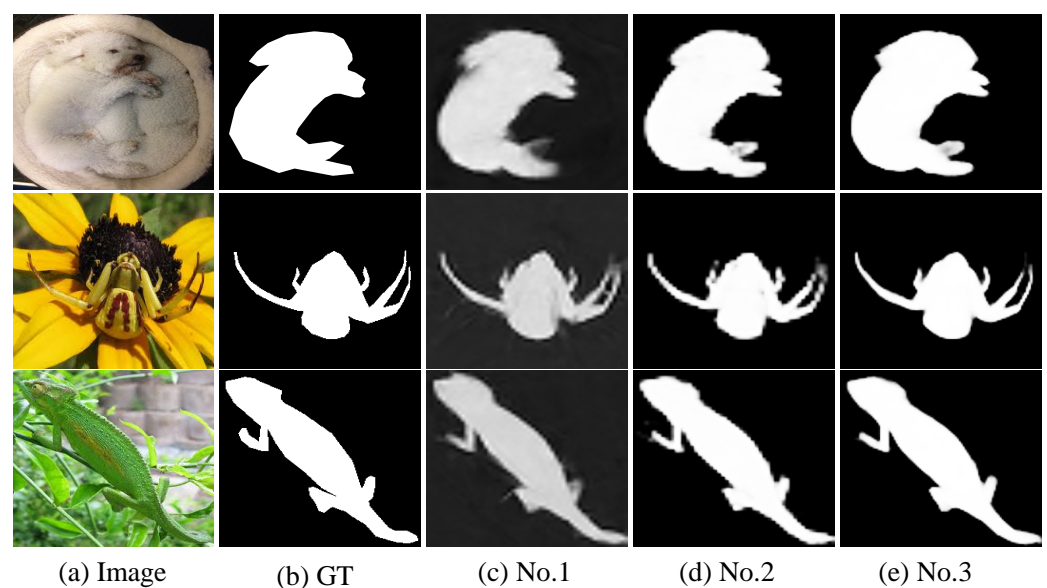


**Figure 5.** Visual comparison of the proposed model with five state-of-the-art COD methods. (The images are from the CAMO [23] dataset).

### 4.3. Ablation Study

To validate the effectiveness of each component and hyperparameter ( $\lambda$ ) in our design, we conducted ablation experiments on three publicly available datasets. The results of these experiments are summarized in Tables 2–4.

**Effectiveness of AFFM.** From Table 2, we observe that Model 5 outperforms Model 4. Specifically, utilizing camouflage cues generated by AFFM in Model 5 resulted in an average improvement of 0.5% in the  $S_\alpha$  metric, 1.33% in the  $E_\phi$  metric, and 1.23% in the  $F_\beta^w$  metric. Moreover, as depicted in Figure 6, the third column of results illustrates a comprehensive overview of the overall structure of the camouflaged object. This observation suggests that the camouflage cues provided by AFFM indeed contribute to a thorough understanding of the structure, aiding the model in distinguishing the camouflaged object from a similar background. Additionally, these cues furnish ample information for subsequent modules to filter and learn more precise details.



**Figure 6.** Visual comparisons for showing the benefits of different modules. No.1, No.2, and No.3 depict the predictions generated by the model following AFFM, FSM, and ERM, respectively. (The images are from the CAMO [23] dataset. The effect images of the FSM and ERM modules are taken from the  $O_1$  branch in Figure 2).

**Effectiveness of FSM.** As shown in Table 2, Model 4 demonstrates significant performance improvements over Model 2 on the three benchmark datasets. Specifically, the average increase in the  $S_\alpha$  indicator was 0.3%, the average increase in the  $E_\phi$  indicator was 0.26%, and the average increase in the  $F_\beta^w$  indicator was 0.8%. Furthermore, as illustrated in Figure 6, in comparison to the predictions in the third column, the predictions in the fourth column include more detailed features, such as the dog's legs in the first row. This observation suggests that the FSM contributes to enhancing the model's comprehension of the overall structure of camouflaged objects as well as mitigating uneven pixel distribution.

**Effectiveness of ERM.** In Table 2, the improved detection accuracy when comparing Model 3 with Model 5, or when comparing Model 1 with Model 2, clearly demonstrates that the inclusion of ERMs significantly enhances the model's ability to detect edge details. This is evident from the average increase of 0.56% in the  $S_\alpha$  metric, 0.3% in the  $E_\phi$  metric, and 0.36% in the  $F_\beta^w$  metric across the three benchmark datasets. In Figure 6, we observe that the predictions in the fifth column exhibit smoother edge features compared to those in the fourth column, as evident in the representation of the lizard abdomen in the third row of predictions. Additionally, the fifth column predictions present more comprehensive predictions for camouflaged objects, such as the legs of spiders in the second row. This

observation implies that the exploration of contextual information between the feature layers and the filtering of features based on the semantic relevance of different branches within the same feature to enhance fine-grained feature representation can effectively enhance the model's performance in detecting the edge information of camouflaged objects.

**Table 2.** Quantitative evaluation for ablation studies on three datasets. The best results are highlighted in bold.

Model	Method	CAMO-Test				COD10K-Test				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
1	Base	0.799	0.860	0.717	0.078	0.801	0.867	0.655	0.038	0.834	0.887	0.745	0.050
2	Base+ERM	0.814	0.852	0.727	0.078	0.827	0.875	0.690	0.034	0.854	0.892	0.771	0.046
3	Base+AFFM+FSM	0.818	0.870	0.746	0.072	0.832	0.885	0.710	<b>0.032</b>	0.852	<b>0.903</b>	0.782	0.045
4	Base+FSM+ERM	0.822	0.864	0.745	0.074	0.827	0.871	0.694	0.035	0.855	0.892	0.773	0.047
5 (ours)	Base+AFFM+FSM+ERM	<b>0.827</b>	<b>0.877</b>	<b>0.754</b>	<b>0.069</b>	<b>0.833</b>	<b>0.888</b>	<b>0.710</b>	0.033	<b>0.859</b>	0.902	<b>0.785</b>	<b>0.045</b>

**Table 3.** Sensitivity analysis on  $\lambda$ . We compared the best scale factors using the four widely used indicators on three datasets. The best results are highlighted in bold.

Model	$\lambda$	CAMO-Test				COD10K-Test				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
1	0.2	0.817	0.863	0.740	0.075	0.830	0.877	0.705	0.033	0.855	0.898	0.781	0.045
2	0.3	0.826	0.875	0.753	0.070	0.832	0.884	0.703	0.032	0.857	0.899	0.785	<b>0.044</b>
3	0.4	0.825	0.869	0.750	0.070	0.832	0.885	0.708	<b>0.032</b>	0.855	0.897	0.782	0.045
4	0.5	<b>0.827</b>	<b>0.877</b>	<b>0.754</b>	<b>0.069</b>	<b>0.833</b>	<b>0.888</b>	<b>0.710</b>	0.033	<b>0.859</b>	<b>0.902</b>	<b>0.785</b>	0.045
5	0.6	0.815	0.865	0.738	0.076	0.832	0.888	0.706	0.033	0.857	0.901	0.784	0.045

**Sensitivity analysis on  $\lambda$ .** Analyzing the prediction results of the COD10K-test dataset in Table 3, we find that the model's prediction performance shows an increasing and then decreasing trend as the parameter  $\lambda$  increases. Specifically, when  $\lambda$  is less than or equal to 0.4, all the predictors show an increasing trend. When  $\lambda$  is greater than 0.5, all the predictors show a decreasing trend, and the predicted value reaches a peak when  $\lambda$  is at 0.5. When  $\lambda = 0.5$ , the  $S_\alpha$  metric increases by 0.1%, the  $E_\phi$  metric increases by 0.3%, and the  $F_\beta^w$  metric increases by 0.2% compared to the predicted value when  $\lambda = 0.4$  (i.e., the second-highest value in the test). Based on these results, we have set the value of  $\lambda$  to 0.5 in this experiment.

**Table 4.** AFFM adaptive feature fusion experiment. The best results are highlighted in bold.

Model	Layers	CAMO-Test				COD10K-Test				NC4K			
		$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$
1	$f_4$	0.822	0.864	0.745	0.074	0.827	0.871	0.694	0.035	0.855	0.892	0.773	0.047
2	$f_4, f_1$	<b>0.827</b>	0.870	0.752	0.072	0.830	0.877	0.702	0.034	<b>0.859</b>	0.899	0.783	0.045
3	$f_4, f_3, f_1$	0.819	0.866	0.747	0.073	0.831	0.878	0.706	<b>0.032</b>	0.857	0.896	0.782	<b>0.042</b>
4 (Ours)	$f_4, f_3, f_2, f_1$	<b>0.827</b>	<b>0.877</b>	<b>0.754</b>	<b>0.069</b>	<b>0.833</b>	<b>0.888</b>	<b>0.710</b>	0.033	<b>0.859</b>	0.902	<b>0.785</b>	0.045

**AFFM fusion feature layer experiment.** In generating camouflage cues in AFFM, experiments were conducted to ascertain which feature layer information should be combined adaptively. As demonstrated in Table 4, the model's detection performance exhibited a significant improvement by including more feature layers in the fusion process. The most favorable detection outcomes were achieved when all feature layers were employed for adaptive fusion. These findings underscore the essential role played by each feature layer in augmenting the model's comprehension of camouflaged objects. It is crucial to emphasize that the contribution of each layer to the final test result is distinct and irreplaceable.

## 5. Conclusions

In this paper, we propose a novel network, CCGNet, for camouflaged object detection, which does not require additional prior guidance training. We propose a feature fusion module that adaptively fuses multilayer features to generate camouflage cues. We integrate the camouflage cues with a foreground separation module, which filters the fused features to separate the object from the background. Finally, the edge information of the camouflaged object is refined by fusing the context information through the edge refinement module. Our model outperforms other state-of-the-art methods through comprehensive experiments on three benchmark camouflage datasets.

**Author Contributions:** Conceptualization, Y.D.; data curation, Y.D.; formal analysis, Y.D.; investigation, Y.D.; methodology, Y.D.; resources, Y.D.; software, Y.D.; supervision, H.Z., C.L., J.X., Y.X. and Z.L.; validation, Y.D.; visualization, Y.D.; writing—original draft, Y.D.; writing—review and editing, Y.D., H.Z., C.L., J.X., Y.X. and Z.L.; project administration, Y.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: [CAMO: [https://drive.google.com/open?id=1h-OqZdwkuPhBvGcVAwmh0f1NGqlH\\_4B6](https://drive.google.com/open?id=1h-OqZdwkuPhBvGcVAwmh0f1NGqlH_4B6), COD10K: <https://drive.google.com/file/d/1vRYAie0JcNStcSwagmCq55eirGyMYGm5/view>, NC4K: [https://drive.google.com/file/d/1kzpX\\_U3gbgO9MuuwZIWuRVpiB7V6yrAQ/view?usp=sharing](https://drive.google.com/file/d/1kzpX_U3gbgO9MuuwZIWuRVpiB7V6yrAQ/view?usp=sharing)].

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CCGNet	Camouflage Cues Guidance Network
COD	Camouflaged Object Detection
CC	Camouflage Cues
AFFM	Adaptive Feature Fusion Module
FSM	Foreground Separation Module
ERM	Edge Refinement Module
DLA	Deep Layer Attention
BN	Batch Normalization
SiLU	Sigmoid Linear Unit
CBAM	Convolutional Block Attention Module
TEM	Texture Enhancement Module
PPO	Preprocessing Operation
GT	Ground Truth

## References

1. Lu, K.; Qian, Z.; Zhu, J.; Wang, M. Cascaded object detection networks for FMCW radars. *Signal Image Video Process.* **2021**, *15*, 1731–1738. [[CrossRef](#)]
2. Mukilan, P.; Semunigus, W. Human and object detection using hybrid deep convolutional neural network. *Signal Image Video Process.* **2022**, *16*, 1913–1923. [[CrossRef](#)]
3. Pan, T.S.; Huang, H.C.; Lee, J.C.; Chen, C.H. Multi-scale ResNet for real-time underwater object detection. *Signal Image Video Process.* **2021**, *15*, 941–949. [[CrossRef](#)]
4. Liu, J.J.; Hou, Q.; Cheng, M.M.; Feng, J.; Jiang, J. A simple pooling-based design for real-time salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3917–3926.
5. Yang, F.; Zhai, Q.; Li, X.; Huang, R.; Luo, A.; Cheng, H.; Fan, D.P. Uncertainty-guided transformer reasoning for camouflaged object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 4146–4155.

6. He, C.; Li, K.; Zhang, Y.; Tang, L.; Zhang, Y.; Guo, Z.; Li, X. Camouflaged object detection with feature decomposition and edge reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 22046–22055.
7. Fan, D.P.; Ji, G.P.; Cheng, M.M.; Shao, L. Concealed object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6024–6042. [[CrossRef](#)] [[PubMed](#)]
8. Troscianko, T.; Benton, C.P.; Lovell, P.G.; Tolhurst, D.J.; Pizlo, Z. Camouflage and visual perception. *Philos. Trans. R. Soc. B Biol. Sci.* **2009**, *364*, 449–461. [[CrossRef](#)]
9. Cuthill, I.C.; Stevens, M.; Sheppard, J.; Maddocks, T.; Párraga, C.A.; Troscianko, T.S. Disruptive coloration and background pattern matching. *Nature* **2005**, *434*, 72–74. [[CrossRef](#)] [[PubMed](#)]
10. Zhang, M.; Xu, S.; Piao, Y.; Shi, D.; Lin, S.; Lu, H. Preynet: Preying on camouflaged objects. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 5323–5332.
11. Zheng, Y.; Zhang, X.; Wang, F.; Cao, T.; Sun, M.; Wang, X. Detection of people with camouflage pattern via dense deconvolution network. *IEEE Signal Process. Lett.* **2018**, *26*, 29–33. [[CrossRef](#)]
12. Fang, Z.; Zhang, X.; Deng, X.; Cao, T.; Zheng, C. Camouflage people detection via strong semantic dilation network. In Proceedings of the ACM Turing Celebration Conference-China, Chengdu, China, 17–19 May 2019; pp. 1–7.
13. Chu, H.K.; Hsu, W.H.; Mitra, N.J.; Cohen-Or, D.; Wong, T.T.; Lee, T.Y. Camouflage images. *ACM Trans. Graph.* **2010**, *29*, 51. [[CrossRef](#)]
14. Xie, E.; Wang, W.; Wang, W.; Ding, M.; Shen, C.; Luo, P. Segmenting transparent objects in the wild. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIII 16, pp. 696–711.
15. Fan, D.P.; Ji, G.P.; Zhou, T.; Chen, G.; Fu, H.; Shen, J.; Shao, L. Pranet: Parallel reverse attention network for polyp segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Lima, Peru, 4–8 October 2020; pp. 263–273.
16. Liu, J.; Zhang, J.; Barnes, N. Modeling aleatoric uncertainty for camouflaged object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 1445–1454.
17. Ji, G.P.; Zhu, L.; Zhuge, M.; Fu, K. Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognit.* **2022**, *123*, 108414. [[CrossRef](#)]
18. Lv, Y.; Zhang, J.; Dai, Y.; Li, A.; Barnes, N.; Fan, D.P. Towards deeper understanding of camouflaged object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *33*, 3462–3476. [[CrossRef](#)]
19. Fan, D.P.; Ji, G.P.; Sun, G.; Cheng, M.M.; Shen, J.; Shao, L. Camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 2777–2787.
20. Sun, Y.; Wang, S.; Chen, C.; Xiang, T.Z. Boundary-guided camouflaged object detection. *arXiv* **2022**, arXiv:2207.00794.
21. Zhai, Q.; Li, X.; Yang, F.; Chen, C.; Cheng, H.; Fan, D.P. Mutual graph learning for camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 12997–13007.
22. Lv, Y.; Zhang, J.; Dai, Y.; Li, A.; Liu, B.; Barnes, N.; Fan, D.P. Simultaneously localize, segment and rank the camouflaged objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 11591–11601.
23. Le, T.N.; Nguyen, T.V.; Nie, Z.; Tran, M.T.; Sugimoto, A. Anabran network for camouflaged object segmentation. *Comput. Vis. Image Underst.* **2019**, *184*, 45–56. [[CrossRef](#)]
24. Skurowski, P.; Abdulameer, H.; Błaszczuk, J.; Depta, T.; Kornacki, A.; Kozieł, P. Animal camouflage analysis: Chameleon database. *Unpubl. Manuscr.* **2018**, *2*, 7.
25. He, C.; Xu, L.; Qiu, Z. Eldnet: Establishment and Refinement of Edge Likelihood Distributions for Camouflaged Object Detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 621–625.
26. Kajjura, N.; Liu, H.; Satoh, S. Improving camouflaged object detection with the uncertainty of pseudo-edge labels. In Proceedings of the ACM Multimedia Asia, Gold Coast, Australia, 1–3 December 2021; pp. 1–7.
27. Zhu, J.; Zhang, X.; Zhang, S.; Liu, J. Inferring camouflaged objects by texture-aware interactive guidance network. *AAAI Conf. Artif. Intell.* **2021**, *35*, 3599–3607. [[CrossRef](#)]
28. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [[CrossRef](#)]
29. Li, A.; Zhang, J.; Lv, Y.; Liu, B.; Zhang, T.; Dai, Y. Uncertainty-aware joint salient object and camouflaged object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10071–10081.
30. Feng, M.; Lu, H.; Ding, E. Attentive feedback network for boundary-aware salient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1623–1632.
31. Perazzi, F.; Krähenbühl, P.; Pritch, Y.; Hornung, A. Saliency filters: Contrast based filtering for salient region detection. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 733–740.
32. Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 15908–15919.

33. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [[CrossRef](#)] [[PubMed](#)]
34. Bian, P.; Jin, Y.; Zhang, N.r. Fuzzy c-means clustering based digital camouflage pattern design and its evaluation. In Proceedings of the IEEE 10th International Conference on Signal Processing Proceedings, Beijing, China, 24–28 October 2010; pp. 1017–1020.
35. Song, L.; Geng, W. A new camouflage texture evaluation method based on WSSIM and nature image features. In Proceedings of the 2010 International Conference on Multimedia Technology, Ningbo, China, 29–31 October 2010; pp. 1–4.
36. Mei, H.; Ji, G.P.; Wei, Z.; Yang, X.; Wei, X.; Fan, D.P. Camouflaged object segmentation with distraction mining. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 8772–8781.
37. Sun, Y.; Chen, G.; Zhou, T.; Zhang, Y.; Liu, N. Context-aware cross-level fusion network for camouflaged object detection. *arXiv* **2021**, arXiv:2105.12555.
38. Jia, Q.; Yao, S.; Liu, Y.; Fan, X.; Liu, R.; Luo, Z. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4713–4722.
39. Ren, J.; Hu, X.; Zhu, L.; Xu, X.; Xu, Y.; Wang, W.; Deng, Z.; Heng, P.A. Deep texture-aware features for camouflaged object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *33*, 1157–1167. [[CrossRef](#)]
40. Dong, B.; Zhuge, M.; Wang, Y.; Bi, H.; Chen, G. Accurate camouflaged object detection via mixture convolution and interactive fusion. *arXiv* **2021**, arXiv:2101.05687.
41. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [[CrossRef](#)]
42. Zhao, J.X.; Liu, J.J.; Fan, D.P.; Cao, Y.; Yang, J.; Cheng, M.M. EGNNet: Edge guidance network for salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 8779–8788.
43. Niu, B.; Wen, W.; Ren, W.; Zhang, X.; Yang, L.; Wang, S.; Zhang, K.; Cao, X.; Shen, H. Single image super-resolution via a holistic attention network. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 191–207.
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
45. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11534–11542.
46. Gu, Z.; Cheng, J.; Fu, H.; Zhou, K.; Hao, H.; Zhao, Y.; Zhang, T.; Gao, S.; Liu, J. Ce-net: Context encoder network for 2d medical image segmentation. *IEEE Trans. Med. Imaging* **2019**, *38*, 2281–2292. [[CrossRef](#)]
47. Wei, J.; Wang, S.; Huang, Q. F<sup>3</sup>Net: Fusion, feedback and focus for salient object detection. *AAAI Conf. Artif. Intell.* **2020**, *34*, 12321–12328. [[CrossRef](#)]
48. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
49. Margolin, R.; Zelnik-Manor, L.; Tal, A. How to evaluate foreground maps? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Virtual, 6–12 September 2014; pp. 248–255.
50. Fan, D.P.; Gong, C.; Cao, Y.; Ren, B.; Cheng, M.M.; Borji, A. Enhanced-alignment measure for binary foreground map evaluation. *arXiv* **2018**, arXiv:1805.10421.
51. Fan, D.P.; Cheng, M.M.; Liu, Y.; Li, T.; Borji, A. Structure-measure: A new way to evaluate foreground maps. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4548–4557.
52. Wu, Z.; Su, L.; Huang, Q. Stacked cross refinement network for edge-aware salient object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7264–7273.
53. Gao, S.H.; Tan, Y.Q.; Cheng, M.M.; Lu, C.; Chen, Y.; Yan, S. Highly efficient salient object detection with 100 k parameters. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 702–721.
54. Qin, X.; Fan, D.P.; Huang, C.; Diagne, C.; Zhang, Z.; Sant’Anna, A.C.; Suarez, A.; Jagersand, M.; Shao, L. Boundary-aware segmentation network for mobile and web applications. *arXiv* **2021**, arXiv:2101.04704.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.