

# FINDING THE ACHILLES HEEL: PROGRESSIVE IDENTIFICATION NETWORK FOR CAMOUFLAGED OBJECT DETECTION

*Mu-Chun Chou\**, *Hung-Jen Chen<sup>†</sup>* and *Hong-Han Shuai<sup>‡</sup>*

\*michael861227.eed05@nctu.edu.tw; hjc.eed07g@nctu.edu.tw<sup>†</sup>; and

<sup>‡</sup>hhshuai@nctu.edu.tw.

## ABSTRACT

Camouflaged object detection (COD) aims to segment objects assimilating into their surroundings. The key challenge for COD is that there are existing high intrinsic similarities between the target object and the background. To solve this challenging problem, we propose the Cascaded Decamouflage Module to progressively improve the prediction map, where each decamouflage module is composed of the region enhancement block and the reverse attention mining block to accurately detect the camouflaged object and obtain complete target objects. In addition, we introduce the classification-based label reweighting to produce the gated label maps as the supervision for assisting the network to capture the most conspicuous region of a camouflaged object and obtain the target object entirely. Extensive experiments on three challenging datasets demonstrate that the proposed model outperforms state-of-the-art methods under different evaluation metrics.

**Index Terms**— Camouflaged object detection, label reweighting

## 1. INTRODUCTION

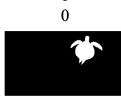
Camouflaged Object Detection (COD) aims at identifying the concealed objects in an image, which has a wide range of valuable applications, e.g., medical diagnosis [1], art [2], and security [3]. There are several early methods [4, 5, 6] that address camouflaged object detection with hand-crafted features like texture, 3D convexity, and motion to detect camouflaged objects. However, such low-level features are designed to segment the most discriminative object in an image, which may not be effective features for detecting camouflaged objects.

Recently, convolutional neural networks (CNNs) have demonstrated powerful capabilities of feature representation. Even if deep learning methods have shown a great performance [7, 8, 9], we observe that existing approaches usually only capture parts of the camouflaged objects. To demonstrate this problem, we calculated the error only for the camouflaged object regions to analyze whether a method fully localizes the camouflaged object. Table 1 shows the results of two state-of-the-art methods on COD10K dataset [9] and

**Table 1.** Compared two SOTA methods with our method by MAE and MAE-CA. All the networks below are using ResNet-50 as the same backbone.

	COD10K [9]		CAMO [8]	
	MAE	MAE-CA	MAE	MAE-CA
SINet.v2 [10]	0.046	0.263	0.085	0.273
PFNet [11]	0.04	0.279	0.085	0.263
Ours	<b>0.033</b>	<b>0.221</b>	<b>0.075</b>	<b>0.224</b>

				
MAE: MAE-CA:	0 0	0.013 0.170	0.008 0.628	0.001 0.126
Image				
MAE: MAE-CA: Image	0 0 GT	0.040 0.574 SINet_v2	0.039 0.583 PFNet	0.018 0.061 Ours

**Fig. 1.** An illustrative example with MAE and MAE-CA.

CAMO dataset [8] in terms of the mean absolute error of camouflaged objects (excluding the background region), abbreviated by MAE-CA. The results shown that although these methods obtain small errors in global prediction, they perform much worse in camouflaged object areas, showing that predicting the whole camouflaged object is still challenging. In addition, Fig. 1 further exhibits two examples, which also manifests that it is challenging for the state-of-the-art methods to detect camouflaged objects completely, something our proposed approach is capable of doing better.

Therefore, we propose the cascaded decamouflage module (CDM) and introduce the classification-based label reweighting (CLR) to better detect camouflaged object. The key idea is to first find the most salient part of the camouflaged objects (Achilles heel) and refine the region progressively. Our contributions are summarized as the following:

- We propose the cascaded decamouflage module to progressively amend and improve the prediction map by refining the predicted camouflaged object areas and

mining the overlooked camouflaged object regions.

- Classification-based label reweighting is proposed to generate the gated label maps for the additional supervision. This helps the network focus more on the noticeable region of a camouflaged object and detect the more target object regions.
- Experimental results on three benchmark datasets demonstrate that the proposed model outperforms the state-of-the-art methods.

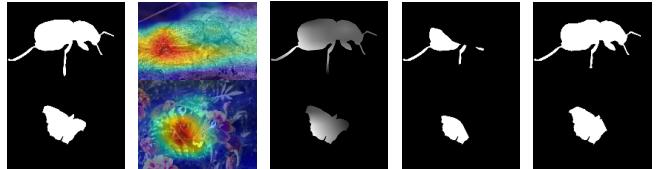
## 2. RELATED WORK

Salient object detection is similar to COD but aims to detect and segment the most attention-grabbing object(s) in an image. Inspired by Fully Convolutional Networks (FCN) [12], different methods are proposed to directly output pixel-wise saliency maps [13, 14, 15]. For example, Chen *et al.* [14] integrated multi-level features and generated the global context information at different stages to learn the relationship among different salient regions. Pang *et al.* [15] integrated similar resolution features of adjacent levels in the encoder to reduce the noise and extracted the multi-scale information from a single level feature for the decoder. Compared with COD, salient object detection usually does not suffer from partially-detected objects since the salient objects are distinguishable from the background.

Recently, several datasets related to camouflaged object detection [7, 8, 9] have been proposed. For example, Le *et al.* [8] proposed an end-to-end network, called ANet, which integrates classification information into segmentation to accurately capture the camouflaged object. Fan *et al.* [9] introduced SINet consisting of two main modules, namely the search module and the identification module, to look for potential target object(s) and identify it. Furthermore, they improved their network, SINet\_v2 [10], with neighbor connection decoder and group-reverse attention. Mei *et al.* [11] presented a distraction mining strategy to refine the segmentation results. However, these methods can not completely predict the whole camouflaged object. As a result, we propose a new framework and a classification-based label reweighting to assist the model from learning conspicuous region to the whole object.

## 3. METHODOLOGY

In this section, we propose the cascaded decamouflage module to progressively refine the prediction map. Then, classification-based label reweighting is introduced to produce the weighted label maps for assisting the network to capture the camouflaged object entirely. The overview of the proposed model is shown in Fig. 3



**Fig. 2.** An illustrative example of classification-based label reweighting (CLR). The second column represents the coarse localization map obtained from GradCAM with the pre-trained classifier. The third column is the weighted label map produced by conducting the multiplication operation between GT and heatmap. The last two columns represents the gated label map with different thresholds.

### 3.1. Feature Extraction and Fusion

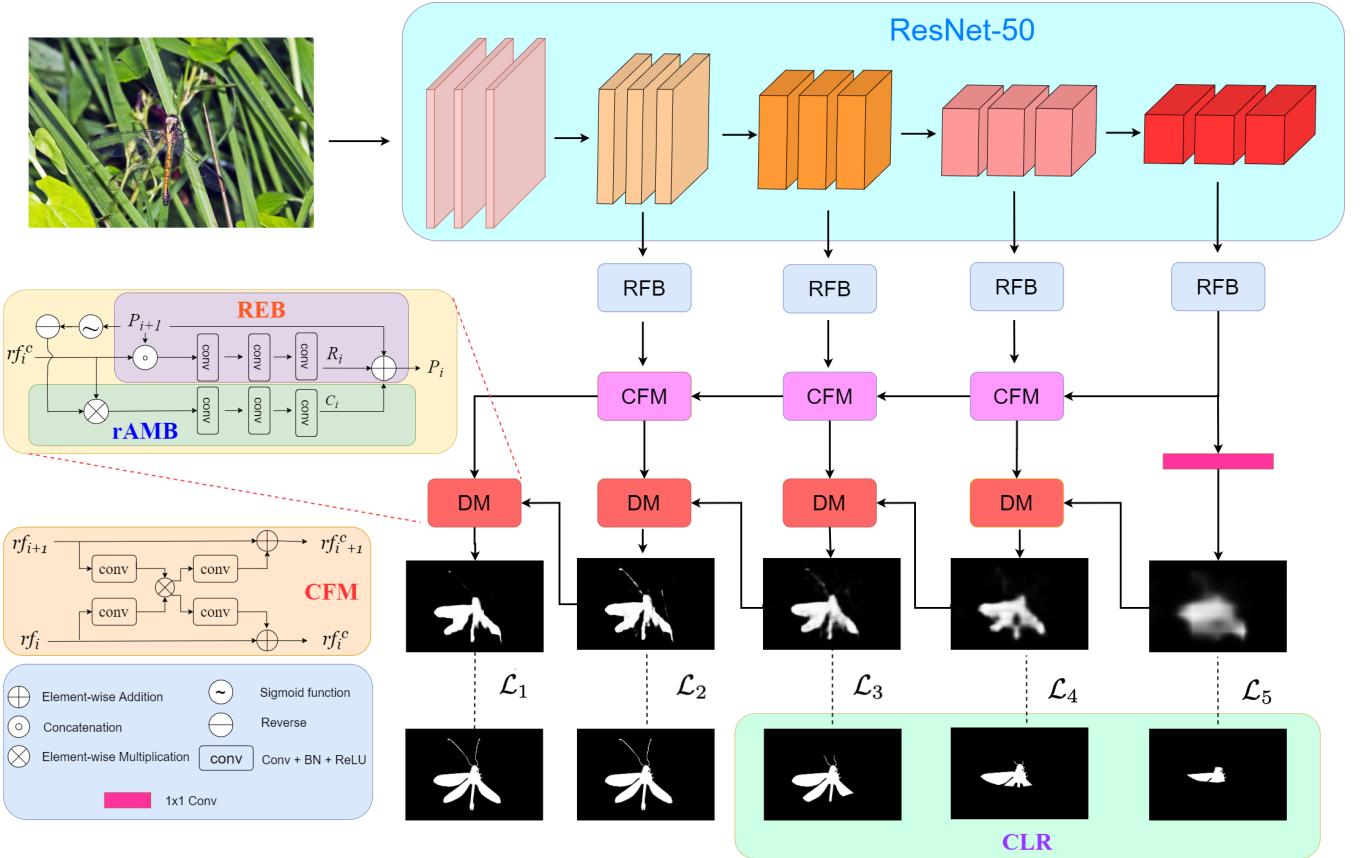
To extract the features, we first leverage ResNet-50 [16] as our backbone network and utilize the Receptive Field (RF) component [17] to incorporate more discriminative feature representations. According to previous work, the low-level features greatly increase computation cost, but bring limited performance improvement. Thus, we only utilize features  $\{f_1, f_2, f_3, f_4\}$ . Afterward, these features are respectively fed into four RF components to enlarge the receptive field. The output features are represented as  $\{rf_1, rf_2, rf_3, rf_4\}$ , and cross feature module (CFM) [18] is then utilized to adaptively select complementary components from input features before fusion, which can effectively avoid introducing too much redundant information. Specifically, given two feature maps  $\{rf_i, rf_{i+1} | i = 1, 2, 3\}$ , these features are transformed through convolutional layers and fused by multiplication to share the properties of both of them. Each CFM is defined as:

$$\begin{aligned} w &= \Phi_i^G(rf_i) \odot \Phi_{i+1}^G(rf_{i+1}), \\ rf_i^c &= rf_i + \Phi_i^G(w), \\ rf_{i+1}^c &= rf_{i+1} + \Phi_{i+1}^G(w), \end{aligned} \quad (1)$$

where  $\Phi^G$  is the combination of convolution, batchnorm, and ReLU activation. We aggregate the features by CFM from high level to low level to obtain the refined features  $\{rf_1^c, rf_2^c, rf_3^c, rf_4^c\}$ .

### 3.2. Cascaded Decamouflage Module

Since multi-level features may contain some redundant parts, we propose the Cascaded Decamouflage Module (CDM) consisting of multiple decamouflage modules (DMs) to progressively amend and refine the output segmentation map by integrating the features of deeper layers. Each decamouflage module (DM) contains a region enhancement block (REB) and a reverse attention mining block (rAMB) to effectively refine the coarse segmentation map. REB improves the prediction map provided by high-level features through integrating the low-level features. However, REB could lose the cor-



**Fig. 3.** An overview of our proposed progressive identification network (PINet). ResNet-50 [16] is used as our backbone feature extractor. The cascaded decamouflage module (CDM) is proposed to progressively amend and refine the coarse prediction map.

rect camouflaged object when the coarse map classifies the true camouflaged objects as background. rAMB is introduced to discover complementary regions and find the overlooked camouflaged object areas by erasing the camouflaged object region which is already detected by high-level features. The erasing strategy driven by reverse attention is proposed to recall the overlooked camouflaged objects and refine the coarse estimation into a complete prediction map.

Specifically, as REB and rAMB take a segmentation map as an input, we take the coarse prediction map  $P_5$  produced from the feature map  $rf_4$  as the initial map. Formally, REB is defined as:

$$R_i = \Phi_i^{REB}(Cat(P_{i+1}, rf_i^c)), \quad (2)$$

where  $i = 1, \dots, 4$ ,  $Cat(\cdot)$  is the concatenation operation,  $\Phi_i^{REB}$  consists of several convolutional layers. rAMB is then defined as:

$$\begin{aligned} rA_i &= 1 - (\sigma(P_{i+1})), \\ C_i &= \Phi_i^{rAMB}(rf_i^c \odot rA_i), \end{aligned} \quad (3)$$

where  $\sigma(\cdot)$  is the Sigmoid function, and  $\odot$  represents the element-wise multiplication. Finally, two output features  $R_i$

and  $C_i$  are added with  $P_{i+1}$  in an element-wise way to obtain the prediction map  $P_i$ .

$$P_i = P_{i+1} + R_i + C_i. \quad (4)$$

While obtaining  $P_i$ , we directly output the prediction map through a sigmoid function. Equipped with the decamouflage module, the network could refine the prediction map by enhancing the detected camouflaged object region and discovering the overlooked object areas.

### 3.3. Classification-Based Label Reweighting

Due to the high similarities between the target object and background, it is hard to detect the whole camouflaged object. Instead of detecting the whole target object directly, we supervise the model for detecting the object from its conspicuous region. Accordingly, we propose to generate weighted label maps from the original label map, as shown in Fig. 2. Specifically, we first train a network to predict the probability of containing camouflaged objects in an image. Then, Grad-CAM [19] is adopted to discover the localization of a camouflaged object. However, the coarse localization map is not

accurate enough to be used for the pseudo label. Thus, we conduct the element-wise multiplication operation between the localization map and the ground-truth map to suppress false-positives. As shown in Fig. 2, given the localization map from GradCAM and the ground-truth map, we obtain different gated label maps by setting different thresholds. Intuitively, as the threshold is lower, the region of the weighted map is larger and closer to the ground-truth map.

Specifically, to obtain the localization map  $M^c \in \mathbb{R}^{W \times H}$  of width  $W$  and height  $H$  for the camouflaged object class  $c$ , the gradient of the score of class  $c$ ,  $y^c$ , with respect to feature map  $F^k$  is calculated. These gradient maps are globally averaged to obtain the weights  $\alpha_k^c$ . Then,  $M^c$  is obtained by a weighted combination of feature maps, *i.e.*,

$$\begin{aligned} \alpha_k^c &= \frac{1}{H \times W} \sum_{i=1}^W \sum_{j=1}^H \frac{\partial y^c}{\partial F_{ij}^k}, \\ M^c &= \text{ReLU} \left( \sum_k \alpha_k^c F^k \right). \end{aligned} \quad (5)$$

We then normalize the pixel values in  $M^c$  by using a linear function  $M^c = \frac{M^c - \min(M^c)}{\max(M^c) - \min(M^c)}$  to rescale the original value to  $[0, 1]$ . To eliminate the false positives, we multiply this map with the ground-truth map  $S = M^c \odot G$  to suppress those misclassified pixels. By setting different thresholds  $t$ , we can obtain different segments of a camouflaged object, which represents the gated label maps  $S^t$ , *i.e.*,

$$S^t(i, j) = \begin{cases} 1, & \text{if } S(i, j) \geq t \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The gated label maps are utilized for additional supervision, which assists the model to detect the conspicuous region and then progressively include more target objects.

### 3.4. Loss Function

Binary cross entropy (BCE) is commonly-used as the loss function, which treats all pixels equally and could be diluted if the background is dominant in an image. In addition, BCE calculates the loss for each pixel independently and ignores the structure of the camouflaged object. To fix these problems, inspired by [18], we utilize the weighted binary cross entropy loss ( $\mathcal{L}_{BCE}^w$ ) and weighted IoU loss ( $\mathcal{L}_{IoU}^w$ ). In addition, we introduce a gated structure-aware loss similar to [27] for reducing the distraction caused by the background as follows,

$$\mathcal{L}_g = \sum_{i,j} \sum_{d \in \vec{x}, \vec{y}} \Psi(|\partial_d P(i, j)| e^{-\alpha |\partial_d(I^g(i, j))|}), \quad (7)$$

where  $\Psi$  is defined as  $\Psi(s) = \sqrt{s^2 + 1e^{-6}}$ ,  $I^g(i, j)$  is the gray-scale image intensity value at  $(i, j)$ ,  $d$  indicates the partial derivatives on the  $\vec{x}$ ,  $\vec{y}$  directions. We adopt deep supervision strategy with the five outputs  $(P_1, P_2, P_3, P_4, P_5)$ .

Each map is up-sampled to the same size as the ground-truth map  $G$ . As such, the total loss can be calculated by:

$$\mathcal{L}_i = \begin{cases} \mathcal{L}_{BCE}^w(P_i, G) + \mathcal{L}_{IoU}^w(P_i, G) + \mathcal{L}_g(P_i, G), & \text{for } i=1, 2, \\ \mathcal{L}_{BCE}^w(P_j \odot S^t, G \odot S^t), & \text{for } i=3, 4, 5, \end{cases} \quad (8)$$

where  $S^t$  is the gated label maps with threshold  $t$ .<sup>1</sup> The total loss is obtained by summarizing all the losses as follows.

$$\mathcal{L}_{total} = \sum_{i=1}^5 \lambda_i \mathcal{L}_i, \quad (9)$$

where  $\lambda_i$  is the weight of each loss for  $\mathcal{L}_i$ .

## 4. EXPERIMENTAL RESULTS

**Dataset.** To evaluate the proposed method, the training sets with 4,640 images (CAMO [8] + COD10K [9] + EXTRA) provided by [9] are used to conduct on the experiment. Three popular benchmark datasets are adopted for testing, including the whole CHAMELEON dataset with 76 images, the test sets of CAMO with 250 images, and COD10K with 2,026 images.

**Evaluation Metric.** Five metrics are utilized to evaluate the performance of our model, *i.e.*,  $S$ -measure ( $S_\alpha$ ),  $E$ -measure ( $E_\phi$ ), weighted  $F$ -measure ( $F_\beta^w$ ), mean absolute error ( $M$ ), and mean absolute error of camouflaged objects (MAE-CA). The MAE and MAE-CA can be calculated as follows:

$$\begin{aligned} MAE &= \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |\mathbf{G}(i, j) - \mathbf{P}(i, j)| \\ MAE-CA &= \frac{1}{R} \sum_{i=1}^W \sum_{j=1}^H |\mathbf{G}(i, j) \cdot \mathbf{P}(i, j) - \mathbf{G}(i, j)| \end{aligned} \quad (10)$$

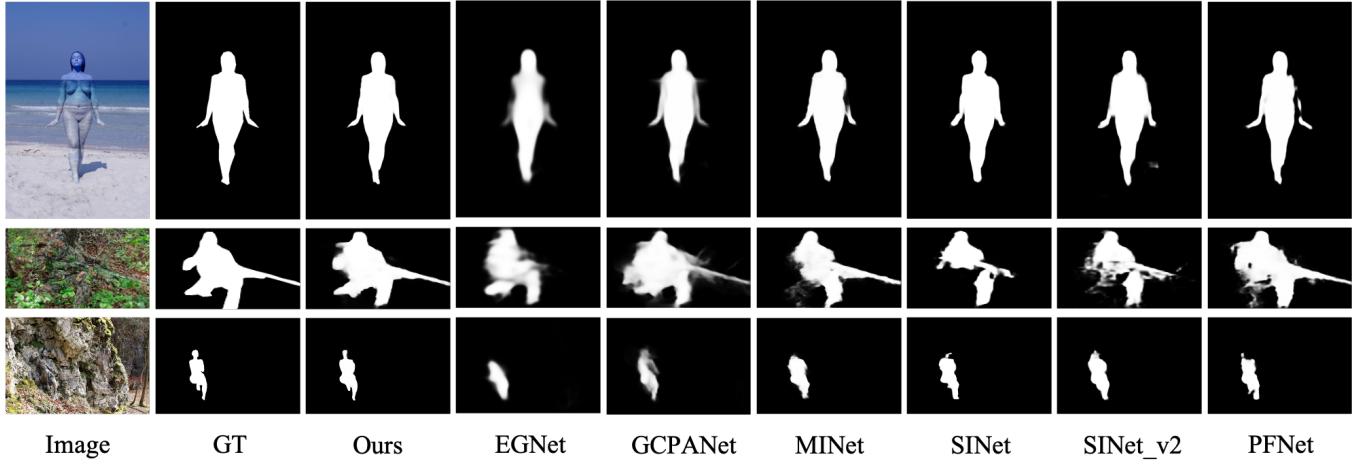
where  $W$  and  $H$  are the width and height of the map, respectively,  $\mathbf{G}_{ij}$  is the label of the pixel  $(i, j)$ ,  $\mathbf{P}_{ij}$  is the probability of being camouflaged objects, and  $R$  is defined as  $\sum_{i=1}^W \sum_{j=1}^H \mathbf{G}(i, j)$ .

**Implementation Details.** For data augmentation, we use horizontal flip, random crop and multi-scale input images. ResNet-50 [16] pre-trained on ImageNet is used as the backbone network. Maximum learning rate is set to 0.001 for the backbone network and 0.01 for other parts. Warm-up and linear decay strategies are used to adjust the learning rate. The whole network is trained end-to-end, with the stochastic gradient descent (SGD) as the optimizer. Momentum and weight decay are set to 0.9 and 0.0005, respectively. Batchsize is set to 36 and maximum epoch is set to 64. The hyperparameters  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$ ,  $\lambda_4$ , and  $\lambda_5$  are respectively set to 1,  $\frac{1}{2}$ ,  $\frac{1}{3}$ ,  $\frac{1}{6}$ , and  $\frac{1}{12}$ . During the inference, each input image is resized to  $544 \times 544$ , and the output map is predicted without any post-processing.

<sup>1</sup>For  $i = 3, 4, 5$ , we respectively set  $t$  to 0.3, 0.5, and 0.7.

**Table 2.** Quantitative results on different datasets. The best results are highlighted in **bold**. Note that  $S_\alpha$ ,  $E_\phi$ ,  $F_\beta^\omega$ ,  $M$ , and MAE-CA denote S-measure, mean E-measure, F-measure, mean absolute error, and mean absolute error of camouflaged objects, respectively. All the networks below are using ResNet-50 as the same backbone.

Baseline Models	CHAMELEON [7]					CAMO [8]					COD10K [9]				
	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	MAE-CA $\downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	MAE-CA $\downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	$F_\beta^\omega \uparrow$	$M \downarrow$	MAE-CA $\downarrow$
BASNet [20]	0.687	0.721	0.474	0.118	0.349	0.618	0.661	0.413	0.159	0.512	0.634	0.678	0.365	0.105	0.426
CPD [21]	0.853	0.866	0.706	0.052	0.235	0.726	0.729	0.550	0.115	0.420	0.747	0.770	0.508	0.059	0.403
HTC [22]	0.517	0.489	0.204	0.129	0.779	0.476	0.442	0.174	0.172	0.832	0.548	0.520	0.221	0.088	0.750
MSRCNN [23]	0.637	0.686	0.443	0.091	0.571	0.617	0.669	0.454	0.133	0.557	0.641	0.706	0.419	0.073	0.522
PFANet [24]	0.679	0.648	0.378	0.144	0.442	0.659	0.622	0.391	0.172	0.488	0.636	0.618	0.286	0.128	0.437
PoolNet [25]	0.776	0.779	0.555	0.081	0.375	0.702	0.698	0.494	0.129	0.466	0.705	0.713	0.416	0.074	0.467
EGNet [26]	0.848	0.870	0.702	0.050	0.260	0.732	0.768	0.583	0.104	0.407	0.737	0.779	0.509	0.056	0.418
GCPANet [14]	0.848	0.851	0.682	0.054	0.241	0.748	0.750	0.578	0.110	0.369	0.764	0.771	0.522	0.058	0.369
MINet [15]	0.852	0.906	0.760	0.038	0.239	0.734	0.772	0.616	0.093	0.401	0.769	0.826	0.603	0.042	0.370
SINet [9]	0.872	0.936	0.806	0.034	0.168	0.745	0.804	0.644	0.092	0.356	0.776	0.864	0.631	0.043	0.317
SINet_v2 [10]	0.869	0.92	0.78	0.04	0.127	0.786	0.839	0.684	0.085	0.263	0.781	0.854	0.615	0.046	0.273
PFNet [11]	0.882	0.931	0.81	0.033	0.15	0.782	0.842	0.695	0.085	0.279	0.800	0.877	0.660	0.040	0.263
Ours	<b>0.897</b>	<b>0.948</b>	<b>0.834</b>	<b>0.027</b>	<b>0.129</b>	<b>0.814</b>	<b>0.868</b>	<b>0.737</b>	<b>0.073</b>	<b>0.221</b>	<b>0.825</b>	<b>0.891</b>	<b>0.704</b>	<b>0.035</b>	<b>0.224</b>



**Fig. 4.** Qualitative results comparing to previous state-of-the-art methods. Each row represents one image and corresponding prediction maps. Each column represents several predictions of one method. Obviously our approach is capable of locating the camouflaged object well and obtaining the complete target objects.

#### 4.1. Comparison with State-of-the-art

**Quantitative Comparisons.** Table 2 compares our method with 12 state-of-the-art models in terms of  $S_\alpha$ ,  $E_\phi$ ,  $F_\beta^\omega$ , MAE ( $M$ ), and MAE-CA. The results manifest that our approach outperforms all state-of-the-art methods on all testing datasets. Although the improved margin in terms of  $M$  is slight on several testing datasets, the proposed approach significantly outperforms state-of-the-art methods in terms of MAE-CA, demonstrating that the proposed CLR and erasing strategy are effective for recall the objects.

**Qualitative Comparisons.** Fig. 4 shows that our model can handle challenging scenarios in the camouflaged object detection. Specifically, our model not only captures the correct camouflaged objects, but also obtains the whole target objects. For example, in first row, other models cannot pre-

dict the woman’s arms clearly, but our method can completely detect it. In the second row, our method can totally find the right foot of the object as compared with others. In the last row, although others can roughly detect the body of the man, they cannot distinguish the head from the rock. In contrast, our method can accurately obtain the whole object.

**Table 3.** The ablation study for different modules in two benchmark datasets.

ResNet-50 CDM CFM CLR	CAMO			COD10K		
	$S_\alpha \uparrow$	$E_\phi \uparrow$	MAE-CA $\downarrow$	$S_\alpha \uparrow$	$E_\phi \uparrow$	MAE-CA $\downarrow$
✓	0.577	0.564	0.595	0.615	0.623	0.577
✓ ✓	0.800	0.856	0.244	0.821	0.889	0.238
✓ ✓ ✓	0.804	0.851	0.244	0.826	0.892	0.230
✓ ✓ ✓ ✓	0.814	0.868	0.221	0.825	0.891	0.224

## 4.2. Ablation Study

To evaluate the different modules, we conduct an ablation study by using ResNet-50 as the backbone network and sequentially adding CDM, CFM and CLR modules. Table 3 shows the performance with different modules on CAMO and COD10K datasets. Compared with the baseline using ResNet-50 only, the model with CDM obtains a large performance gain in all evaluation metrics. It is because CDM could progressively amend and refine the coarse prediction map. In addition, CFM makes good use of feature fusion and allows our CDM to perform well in all evaluation metrics on COD10K dataset. The last row demonstrates the effectiveness of the gated label maps produced by CLR. Although the S-measure and E-measure are almost same on COD10K, it makes improvements in MAE-CA because the model with CLR detects the camouflaged objects from the most conspicuous region of them and detects the target objects completely.

## 5. CONCLUSIONS

In this paper, to detect camouflaged objects accurately and completely, we design the cascaded decamouflage module consisting of region enhancement blocks (REB) and reverse attention mining blocks (rAMB) to progressively refine the prediction. In addition, we introduce classification-based label reweighting (CLR) to assist the network to detect the conspicuous parts of camouflaged object and to obtain camouflaged objects completely. In the future, we plan to leverage the contrastive learning for better capturing the patterns of camouflaged objects and reduce the amount of required data.

## 6. REFERENCES

- [1] Deng-Ping Fan et al., “Inf-net: Automatic covid-19 lung infection segmentation from ct images,” *IEEE Transactions on Medical Imaging*, 2020.
- [2] Martin Stevens and Sami Merilaita, “Animal camouflage: current issues and new perspectives,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1516, pp. 423–427, 2009.
- [3] Xu Zhao et al., “Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network,” in *IEEE ICME*, 2017, pp. 343–348.
- [4] P Sengottuvelan, Amitabh Wahi, and A Shanmugam, “Performance of decamouflaging through exploratory image analysis,” in *IEEE ICETET*, 2008, pp. 6–10.
- [5] Yuxin Pan et al., “Study on the camouflaged target detection method based on 3d convexity,” *Modern Applied Science*, vol. 5, no. 4, pp. 152, 2011.
- [6] Jianqin Yin Yanbin Han Wendi Hou and Jinping Li, “Detection of the mobile object with camouflage color under dynamic background based on optical flow,” *Procedia Engineering*, vol. 15, pp. 2201–2205, 2011.
- [7] P Skurowski et al., “Animal camouflage analysis: Chameleon database,” *Unpublished Manuscript*, 2018.
- [8] Trung-Nghia Le et al., “Anabanch network for camouflaged object segmentation,” *Computer Vision and Image Understanding*, vol. 184, pp. 45–56, 2019.
- [9] Deng-Ping Fan et al., “Camouflaged object detection,” in *IEEE CVPR*, 2020.
- [10] Deng-Ping Fan et al., “Concealed object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [11] Haiyang Mei et al., “Camouflaged object segmentation with distraction mining,” in *IEEE CVPR*, 2021, pp. 8772–8781.
- [12] Jonathan Long, Evan Shelhamer, and Trevor Darrell, “Fully convolutional networks for semantic segmentation,” in *IEEE CVPR*, 2015, pp. 3431–3440.
- [13] Qi Qi et al., “Multi-scale capsule attention-based salient object detection with multi-crossed layer connections,” in *IEEE ICME*, 2019, pp. 1762–1767.
- [14] Zuyao Chen et al., “Global context-aware progressive aggregation network for salient object detection,” in *AAAI*, 2020.
- [15] Youwei Pang et al., “Multi-scale interactive network for salient object detection,” in *IEEE CVPR*, 2020.
- [16] Kaiming He et al., “Deep residual learning for image recognition,” in *IEEE CVPR*, 2016, pp. 770–778.
- [17] Songtao Liu, Di Huang, et al., “Receptive field block net for accurate and fast object detection,” in *ECCV*, 2018, pp. 385–400.
- [18] Qingming Huang Jun Wei, Shuhui Wang, “F3net: Fusion, feedback and focus for salient object detection,” in *AAAI*, 2020.
- [19] Ramprasaath R Selvaraju et al., “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *IEEE CVPR*, 2017, pp. 618–626.
- [20] Xuebin Qin et al., “Basnet: Boundary-aware salient object detection,” in *IEEE CVPR*, 2019.
- [21] Zhe Wu, Li Su, and Qingming Huang, “Cascaded partial decoder for fast and accurate salient object detection,” in *IEEE CVPR*, 2019.
- [22] Kai Chen et al., “Hybrid task cascade for instance segmentation,” in *IEEE CVPR*, 2019.
- [23] Zhaojin Huang et al., “Mask scoring r-cnn,” in *IEEE CVPR*, 2019.
- [24] Ting Zhao and Xiangqian Wu, “Pyramid feature attention network for saliency detection,” in *IEEE CVPR*, 2019.
- [25] Jiang-Jiang Liu et al., “A simple pooling-based design for real-time salient object detection,” in *IEEE CVPR*, 2019.
- [26] Jia-Xing Zhao et al., “Egnet: Edge guidance network for salient object detection,” in *IEEE ICCV*, 2019.
- [27] Jing Zhang et al., “Weakly-supervised salient object detection via scribble annotations,” in *IEEE CVPR*, 2020.