Mask-and-Edge Co-Guided Separable Network for Camouflaged Object Detection

Jiesheng Wu, *Student Member, IEEE*, Weiyun Liang, *Graduate Student Member, IEEE*, Fangwei Hao, and Jing Xu, *Member, IEEE*

***Abstract*—Camouflaged object detection (COD) involves segmenting objects that share similar patterns, such as color and texture, with their surroundings. Current methods typically employ multiple well-designed modules or rely on edge cues to learn object feature representations for COD. However, these methods still struggle to capture the discriminative semantics between camouflaged objects (foreground) and background, possibly generating blurry prediction maps. To address these limitations, we propose a novel mask-and-edge co-guided separable network (MECS-Net) for COD that leverages both edge and mask cues to learn more discriminative representations and improve detection performance. Specifically, we design a mask-and-edge co-guided separable attention (MECSA) module, which consists of three flows for separately capturing edge, foreground, and background semantics. In addition, we propose a multi-scale enhancement fusion (MEF) module to aggregate multi-scale features of objects. The predictions are decoded in a top-down manner. Extensive experiments and visualizations demonstrate that our CNN-based and Transformer-based MECS-Net outperform 13 state-of-the-art methods on four popular COD datasets.**

***Index Terms*—Camouflaged object detection, mask-and-edge co-guided, multi-scale enhancement fusion.**

I. Introduction

CAMOUFLAGED object detection (COD) is the task of accurately segmenting foreground objects that blend in with their surroundings due to similar patterns, such as color and texture, but differ semantically. Thus, COD has become a new challenging research [1], [2], which motivates more COD-related studies [3], [4], [5], [6].

Due to the high degree of visual similarity between the foreground and background in camouflaged scenes, COD is a more challenging task compared to other object detection tasks [7] (e.g., salient object detection (SOD) [8]). Early researchers tackled this challenge using hand-crafted features for detection [9], [10], while more recent state-of-the-art (SOTA) methods employ deep learning [1], [2], [11], [12], [13]. For example, SINet [2] draws inspiration from biology [14], proposes two search and recognition modules for locating objects, and contributes a large-scale camouflaged object dataset, namely COD10 K, to promote COD development. Generally, representative COD methods can be categorized into three groups: 1) methods that use elaborated modules or components [15], [16], [17], [18], [19], 2) methods that incorporate additional tasks (e.g., classification [11], SOD [20], and ranking [21]) to enhance camouflaged feature representations, and 3) methods that leverage prior knowledge guidance (e.g., edge cues [13], [22], [23], frequency cues [24], [25], and texture cues [26], [27], [28]) to facilitate object detection.

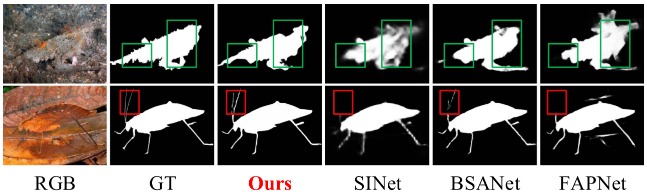

Fig. 1. Visual examples of challenging camouflage scenarios are presented to compare our method with recent state-of-the-art (SOTA) methods [1], [18], [22]. Compared to other methods, our method can learn more ***edge semantics*** (framed by the red boxes) and generate ***clearer*** predicted maps (framed by the green boxes).

Although these methods achieve significant improvements for COD, some potential issues are still worth investigating. As shown in Fig. 1, we observe that edge-guided based methods (BSANet [22], FAPNet [18]) can predict more edge details compared to SINet [1], which indicates that edge cues can provide more object-related edge semantics and locate objects accurately (see the 1^{st} row of Fig. 1). However, only edge-guided methods may obtain blurry prediction maps (see the 2^{nd} row of Fig. 1). One empirical explanation for this observation is that relying solely on edge guidance while effective in localizing objects, may not adequately capture subtle discriminative semantics due to the high similarity between foreground and background. The key solution is to learn representations for foreground and background separately.

Motivated by the insights above, we propose a mask-and-edge co-guided separable network (MECS-Net) for COD that simultaneously leverages edge and mask cues to improve performance. Specifically, we design a mask-and-edge co-guided separable attention (MECSA) module to capture edge, foreground, and background semantics separately, promoting MECS-Net to learn more discriminative features between foreground and background. MECSA consists of three flows: a foreground flow, a background flow, and an edge flow. The edge flow is used

Manuscript received 26 February 2023; revised 11 May 2023; accepted 5 June 2023. Date of publication 26 June 2023; date of current version 29 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62002177 and in part by the Tianjin Natural Science Foundation of China under Grants 21JCYBJC00110 and 19JCQNJC00300. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Le Lu. *(Jiesheng Wu and Weiyun Liang contributed equally to this work.) (Corresponding author: Jing Xu.)*

The authors are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China (e-mail: jasonwu@mail.nankai.edu.cn; weiyunliang@mail.nankai.edu.cn; haofangwei@mail.nankai.edu.cn; xujing@nankai.edu.cn).

Codes and results are available https://github.com/TomorrowJW/MECS-Net-COD∗.

Digital Object Identifier 10.1109/LSP.2023.3286787

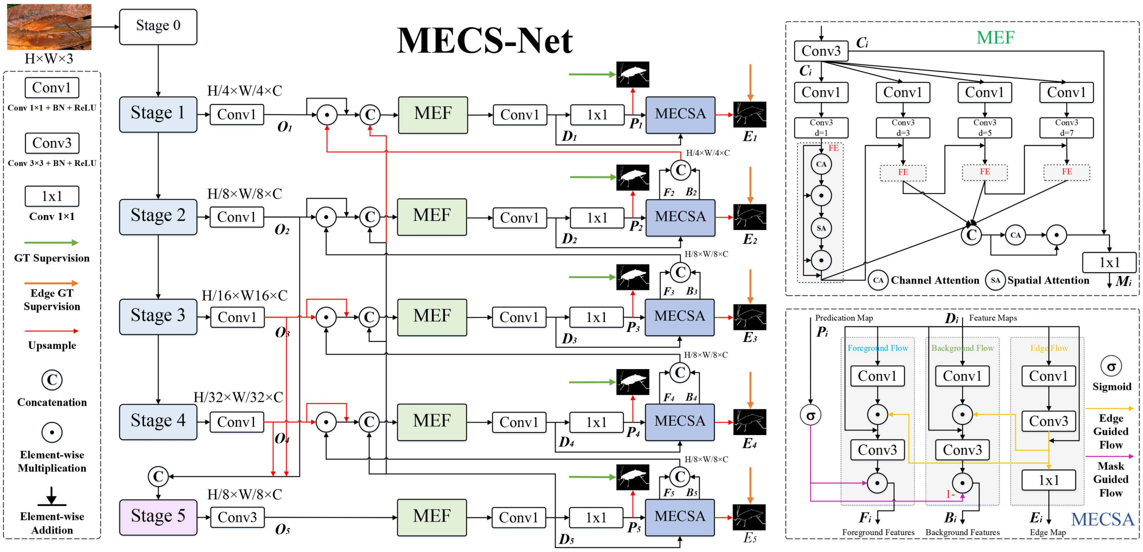

Fig. 2. Overall architecture of our proposed MECS-Net. MECS-Net mainly consists of three parts: encoder (Res2Net is the default in the work), multi-scale enhancement fusion (MEF) module, and mask-and-edge co-guided separable attention (MECSA) module.

for edge guidance and map generation, while the foreground and background flows are incorporated with masks to generate corresponding foreground and background features. Masks are obtained from the prediction maps output by MECS-Net. In addition, we propose a multi-scale enhancement fusion (MEF) module to capture multi-scale features of objects. Unlike previous multi-scale modules [1], [22], we design a feature enhancement (FE) component in MEF to further boost multi-scale representations. Furthermore, we use multi-level features to generate richer context features for COD. Our main contributions can be summarized as follows:

- We propose a mask-and-edge co-guided separable attention (MECSA) module, which introduces mask and edge cues to capture edge, foreground, and background semantics separately. MECSA promotes MECS-Net to learn edge semantics, which is essential for object localization, as well as discriminative semantics for object detection.
- We present a multi-scale enhancement fusion (MEF) module to learn and enhance multi-scale feature representations for boosting detection performance.
- Extensive experiments demonstrate that our proposed CNN-based and Transformer-based MECS-Net outperform 13 state-of-the-art COD methods across four public datasets in terms of detection performance, edge details, object localization, and prediction clarity.

II. Method

A. Encoder

We present our proposed MECS-Net in Fig. 2. Specifically, for an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, it is first fed into an encoder to generate multi-level features $\{\mathbf{O}_i\}_{i=1}^{4}$, and these features are all accompanied by 1×1 convolutional layers to unify the number of channels. Here, we exclude the 0^{th} level features as they contain more redundant information. Instead, we utilize features from the 1^{st} to 4^{th} levels of the encoder. Then, inspired by [2] and [29], we further concatenate the features of the top three levels to generate higher-level features $\mathbf{O}_5$ by a 3×3 convolutional layer. To this end, we can obtain multi-level features $\{\mathbf{O}_i\}_{i=1}^{5}$.

B. Decoder

As shown in Fig. 2, the entire decoding process of MECS-Net is from top to down. To be specific, $\mathbf{O}_i$ is first fed into our proposed MEF module that can capture multi-scale information to generate contextual features $\mathbf{M}_i$. Then, the obtained features are fed into two successive 1×1 convolutional layers to output the current predicted feature maps $\mathbf{D}_i$ and binary map $\mathbf{P}_i$. Finally, both $\mathbf{D}_i$ and $\mathbf{P}_i$ are fed into our proposed MECSA module to separately learn foreground features $\mathbf{F}_i$ and background features $\mathbf{B}_i$ and output the predicted binary edge map $\mathbf{E}_i$. Furthermore, $\mathbf{D}_5$ participates in the feature decoding process of each level, providing rich semantics for each level. Therefore, these multi-level features $\{\mathbf{O}_i\}_{i=1}^{4}$ are decoded in a progressive manner which can be denoted as follows:

$$\begin{cases}\mathbf{D}_i = \mathrm{Conv1}\left(\mathrm{MEF}\left(\left[\mathbf{O}_i \oplus \left(\mathbf{O}_i \odot \left(\left[\mathbf{F}_{i+1}, \mathbf{B}_{i+1}\right]\right), \mathbf{D}_5\right]\right)\right)\right), \\ \mathbf{P}_i = \mathrm{Conv}_{1\times 1}\left(\mathbf{D}_i\right), \end{cases} \tag{1}$$

where $[.,.]$ denotes the concatenation operation. $\mathrm{Conv1}(\cdot)$ and $\mathrm{Conv}_{1\times1}(\cdot)$ are 1×1 convolutional layers with/without batch normalization and ReLU activation, respectively. $\oplus$ and $\odot$ denote element-wise addition and element-wise multiplication operations, respectively. $\mathrm{MEF}(\cdot)$ denotes the MEF module. Note that in the first, third, and fourth levels, the corresponding features are up-sampled for dimension matching (as shown in the red arrow line in Fig. 2).

C. Multi-Scale Enhancement Fusion Module

We draw inspiration from the works of Fan et al. [2], Sun et al. [23], and Zhu et al. [22], which suggest that multi-scale contextual semantics can significantly enhance the performance of COD. Therefore, we propose a multi-scale enhancement fusion (MEF) module for capturing multi-scale semantics. The upper right corner of Fig. 2 shows the details of the MEF module. Specifically, given an input, we first employ a 3×3 convolutional layer to generate features $\mathbf{C}_i$. Then, $\mathbf{C}_i$ are fed into four parallel 1×1 convolutional layers to generate four different features. Next, these features are fed into four different dilated convolutional layers ($d \in \{1, 3, 5, 7\}$ denotes the corresponding dilation rate) to output multi-scale features $\mathbf{C}_i^j, j \in \{1, 2, 3, 4\}$. Next, $\mathbf{C}_i^j$ are fed into the feature enhancement (FE) component to refine the $\mathbf{C}_i^j$. The forward process of FE can be expressed as

$$\mathbf{M}_i^j = \mathbf{C}_i^j \oplus \left(\mathbf{C}_i^j \odot \mathrm{SA}\left(\mathbf{C}_i^j \odot \mathrm{CA}\left(\mathbf{C}_i^j\right)\right)\right), \tag{2}$$

where $\mathrm{CA}(\cdot)$ and $\mathrm{SA}(\cdot)$ represent the channel attention (CA) [30] and spatial attention (SA) [30], respectively. It is worth noting that for the $\mathbf{C}_i^j, j \in \{2, 3, 4\}$, $\mathbf{M}_i^{j-1}$ are added to the current $\mathbf{C}_i^j$ to obtain new features $\mathbf{C}_i^j$, which integrates the semantics of the previous scale. Finally, all multi-scale enhancement features are concatenated and fused to output the desired features $\mathbf{M}_i$. The entire process be formulated as

$$\begin{cases} \mathbf{N}_i = \left[\mathbf{M}_i^1, \mathbf{M}_i^2, \mathbf{M}_i^3, \mathbf{M}_i^4\right], \\ \mathbf{M}_i = \mathrm{Conv}_{1\times1}\left(\mathbf{C}_i \oplus \left(\mathbf{N}_i \odot \mathrm{CA}\left(\mathbf{N}_i\right)\right)\right). \end{cases} \tag{3}$$

D. Mask-and-Edge Co-Guided Separable Attention Module

Due to their similar patterns to the backgrounds, camouflaged objects pose a challenge for distinguishing foreground and background features. To address this challenge, we propose a novel mask-and-edge co-guided separable attention (MECSA) module that promotes the MECS-Net to learn discriminative features between foreground and background and improve the performance of COD. The lower right corner of Fig. 2 shows the details of the MECSA module. The MECSA module contains foreground, background, and edge flows to learn foreground, background, and edge features, respectively. Specifically, given the predicted features $\mathbf{D}_i$ and binary map $\mathbf{P}_i$, we first use three parallel 1×1 convolutional layers to process the features $\mathbf{D}_i$. Then, for the edge flow, we further use a 3×3 convolutional layer and a 1×1 convolutional layer to obtain the edge features $\mathbf{E}_f$ and edge binary map $\mathbf{E}_i$:

$$\begin{cases} \mathbf{E}_f = \mathbf{D}_i \oplus \mathrm{Conv3}\left(\mathrm{Conv1}\left(\mathbf{D}_i\right)\right), \\ \mathbf{E}_i = \mathrm{Conv}_{1\times1}\left(\mathbf{E}_f\right). \end{cases} \tag{4}$$

For the foreground and background flows, two predicted masks are cooperated with both flows to learn foreground and background features. Thus, we use the predicted binary map $\mathbf{P}_i$ followed with a Sigmoid activation to generate the foreground mask $\mathbf{F}_m$ and background mask $\mathbf{B}_m$ ($\mathbf{B}_m = \mathbf{1} - \mathbf{F}_m$). Finally, we obtain the foreground features $\mathbf{F}_i$ and background features $\mathbf{B}_i$:

$$\begin{cases} \mathbf{F}_i = \mathbf{F}_m \odot \mathrm{Conv3}\left(\mathbf{D}_i \oplus \mathbf{E}_f \odot \left(\mathrm{Conv1}\left(\mathbf{D}_i\right)\right)\right), \\ \mathbf{B}_i = \mathbf{B}_m \odot \mathrm{Conv3}\left(\mathbf{D}_i \oplus \mathbf{E}_f \odot \left(\mathrm{Conv1}\left(\mathbf{D}_i\right)\right)\right). \end{cases} \tag{5}$$

E. Loss Function

Our loss function consists of two parts: the prediction loss $\mathcal{L}_p$ and edge loss $\mathcal{L}_e$. The two losses can be formulated as

$$\begin{cases} \mathcal{L}_p = \sum_{i=1}^{5} \mathcal{L}_{wbce}\left(\mathbf{P}_i, \mathbf{G}\right) + \mathcal{L}_{wiou}\left(\mathbf{P}_i, \mathbf{G}\right), \\ \mathcal{L}_e = \sum_{i=1}^{5} \mathcal{L}_{wbce}\left(\mathbf{E}_i, \mathbf{G_e}\right) + \mathcal{L}_{dice}\left(\mathbf{E}_i, \mathbf{G_e}\right), \end{cases} \tag{6}$$

where $\mathcal{L}_{wbce}$, $\mathcal{L}_{wiou}$, and $\mathcal{L}_{dice}$ denote the weighted binary cross-entropy loss, weighted IOU loss, and dice loss [31], respectively. $\mathbf{G}$ is the ground-truth map and $\mathbf{G_e}$ is the edge ground-truth map. Therefore, the total loss is formulated as $\mathcal{L} = \mathcal{L}_p + \mathcal{L}_e$.

III. Experiment

A. Settings

1) Datasets: We evaluate our proposed MECS-Net on four public datasets: CAMO [11], CHAMELEON [32], COD10K [2], and NC4K [21]. Following the same setup [2], 1,000 images from CAMO and 3,040 images from COD10 K are used for training, and all remaining images are used for testing.

2) Metrics: Following previous works [1], [2], four evaluation metrics are used in the COD field, including S-measure (S_α) [33], weighted F-measure (F_β^ω) [34], mean absolute error (M) [35], and mean E-measure (E_ϕ) [36].

3) Implementation Details: We apply the PyTorch library to implement our MECS-Net. Two representative pre-trained backbones Res2Net-50 [37] and Swin Transformer V2 [38], [39] (Swin V2) are used as our encoders. During the training process, all input images are resized to 384 × 384. Adam [40] is used as our optimizer, and the mini-batch is set to 18. The initial learning rate is 1e-4 and follows the StepLR decay strategy that is divided by two every 20 epochs. Our model is trained for 200 epochs on 2 Nvidia V100 GPUs (with 32 GB memory). Furthermore, our model performs inference and prediction on the aforementioned GPUs at an average speed of 14.51 FPS.

4) Comparison With State-of-The-Arts: We select 12 recent SOTA methods (CNN-based) for performance comparisons with our method, including EGNet [8], SINet [1], LSR [21], PFNet [15], UGTR [41], ERRNet [13], C ^{2} FNet [17], SINetV2 [2], BSANet [22], DTCNet [26], FAPNet [18], R-MGL_v2 [19]. Moreover, we also select one Transformer-based SOTA method LSR+^{2} [42] for comparison.

B. Performance Comparisons

1) Quantitative Comparisons: Table I shows the results of all methods on the four datasets in terms of four metrics. It can be observed that our MECS-Net outperforms all competitors, except that the MECS-Net-R versions on the CAMO dataset are slightly inferior to SINetV2. The results demonstrate the effectiveness of our MECS-Net. Furthermore, MECS-Net-T achieves better performance than LSR+^{2} with Swin V2-based and significantly outperforms all CNN-based models, which shows the excellent feature extraction ability of Swin V2.

2) Qualitative Comparisons: The visual comparisons of all methods are presented in Fig. 3, which demonstrate that MECS-Net achieves superior visual predictions to other methods in terms of object localization, edge details, and prediction clarity. Specifically, MECS-Net effectively handles various camouflage scenarios by accurately predicting detailed and complete object structures, even in cases where camouflaged objects share high similarity with the background and possess complex edges (see the 1^{st} and 2^{nd} rows of Fig. 3). MECS-Net also adapts to camouflaged objects at different scales and detects objects accurately (see the 3^{rd} and 4^{th} rows of Fig. 3). Moreover, MECS-Net performs better in the presence of multiple camouflaged objects (see the 5^{th} row of Fig. 3).

3) Edge Exploration: To explore the effects of the MECSA module, we show the visual examples of predicted edge maps compared to R-MGL_v2 [19] in Fig. 4. Our visual results demonstrate that MECS-Net outperforms R-MGL_v2 in learning edge features for camouflaged objects, resulting in accurate predictions of the edge details and object structures.

TABLE I
COMPARISONS WITH RECENT SOTAS FOR COD ON FOUR DATASETS IN TERMS OF FOUR METRICS

Methods	Publication	CHAMELEON (76)				CAMO-Test (250)				COD10K-Test (2,026)				NC4K (4,121)			
		$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$
EGNet [8]	ICCV2019	0.797	0.859	0.649	0.065	0.732	0.800	0.604	0.109	0.737	0.810	0.517	0.061	0.777	0.841	0.639	0.075
SINet [1]	CVPR2020	0.869	0.891	0.740	0.044	0.751	0.771	0.606	0.100	0.771	0.806	0.551	0.051	0.808	0.871	0.723	0.058
LSR [21]	CVPR2021	0.890	0.935	0.822	0.030	0.787	0.838	0.696	0.080	0.804	0.880	0.673	0.037	0.839	0.895	0.767	0.048
PFNet [15]	CVPR2021	0.882	0.931	0.810	0.033	0.782	0.841	0.695	0.085	0.800	0.877	0.660	0.040	0.829	0.887	0.745	0.053
UGTR [41]	ICCV2021	0.888	0.911	0.796	0.031	0.785	0.823	0.686	0.086	0.818	0.853	0.667	0.035	0.839	0.874	0.747	0.052
ERRNet [13]	PR2022	0.877	0.927	0.805	0.036	0.761	0.817	0.660	0.088	0.780	0.867	0.629	0.044	-	-	-	-
C^{2}FNet [17]	TCSVT2022	0.893	0.946	0.845	0.028	0.800	0.859	0.730	0.077	0.811	0.891	0.691	0.036	0.840	0.896	0.770	0.048
SINetV2 [2]	TPAMI2022	0.888	0.942	0.816	0.030	**0.820**	**0.882**	**0.743**	**0.070**	0.815	0.887	0.680	0.037	0.847	0.903	0.770	0.048
BSANet [22]	AAAI2022	0.895	0.946	0.841	0.027	0.794	0.851	0.717	0.079	0.818	0.891	0.699	0.034	0.841	0.897	0.771	0.048
DTCNet [26]	TMM2022	0.876	0.897	0.773	0.039	0.778	0.804	0.667	0.084	0.790	0.821	0.616	0.041	-	-	-	-
FAPNet [18]	TIP2022	0.893	0.940	0.825	0.028	0.815	0.865	0.734	0.076	0.822	0.888	0.694	0.036	0.851	0.899	0.774	0.047
R-MGL_v2 [19]	TIP2022	0.892	0.935	0.825	0.029	0.774	0.848	0.684	0.085	0.813	0.882	0.682	0.034	0.831	0.892	0.751	0.051
Ours-R	2023	**0.896**	**0.940**	**0.852**	**0.026**	0.803	0.860	0.741	0.071	**0.827**	**0.899**	**0.724**	**0.031**	**0.848**	**0.903**	**0.790**	**0.044**
LSR+^{2} [42]	TCSVT2023	0.895	0.943	0.849	0.025	**0.854**	**0.924**	**0.839**	**0.049**	0.847	0.924	0.775	0.028	0.870	0.924	0.845	0.036
Ours-T	2023	**0.901**	**0.953**	**0.857**	**0.023**	0.853	0.911	0.811	0.051	**0.854**	**0.919**	**0.770**	**0.026**	**0.874**	**0.924**	**0.829**	**0.036**

The best results are highlighted in bold. "-" denotes the results are not available. "↑" and "↓" mean that the results are better. "-R": Res2Net. "-T": Swin transformer v2.

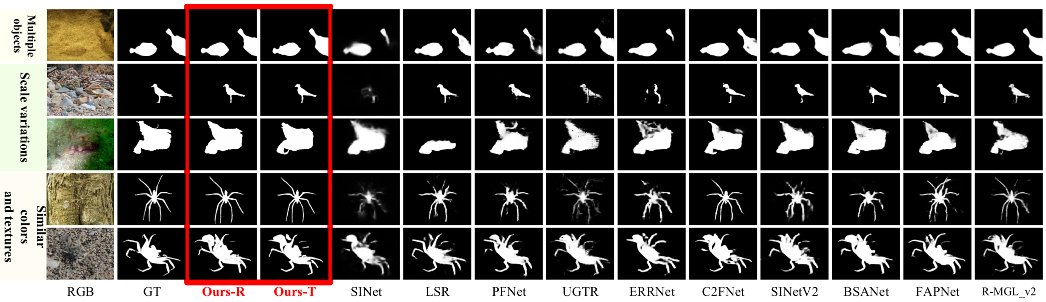

Fig. 3. Visual comparison examples of the MECS-Net with SOTAs.

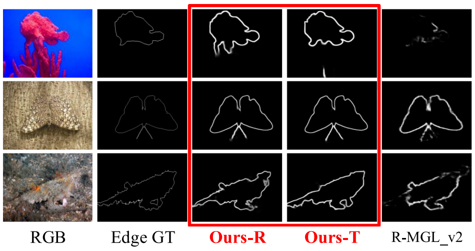

Fig. 4. Edge exploration comparison examples of the MECS-Net with SOTAs.

TABLE II
ABLATION STUDY RESULTS

Methods	MECSA	MEF	E-G	M-G	FE	CAMO-Test (250)				COD10K-Test (2,026)				NC4K (4,121)			
						$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi\uparrow$	$F^\omega_\beta\uparrow$	$M\downarrow$
Baseline						0.793	0.855	0.727	0.076	0.820	0.891	0.709	0.033	0.840	0.896	0.778	0.047
No.1		✓				0.789	0.851	0.719	0.080	0.821	0.895	0.711	0.033	0.842	0.898	0.781	0.048
No.2		✓			✓	0.794	0.852	0.727	0.078	0.821	0.895	0.711	0.033	0.843	0.900	0.781	0.047
No.3	✓		✓	✓		0.794	0.854	0.729	0.076	0.824	0.894	0.717	0.032	0.843	0.898	0.782	0.046
No.4	✓	✓	✓		✓	0.799	0.858	0.735	0.077	0.822	0.895	0.711	0.034	0.844	0.902	0.782	0.047
No.5	✓	✓		✓	✓	0.797	0.858	0.732	0.074	0.826	0.896	0.723	0.032	0.846	0.901	0.786	0.045
ours	✓	✓	✓	✓	✓	**0.803**	**0.860**	**0.741**	**0.071**	**0.827**	**0.899**	**0.724**	**0.030**	**0.848**	**0.903**	**0.790**	**0.044**

"E-G": Edge-Guided Flow. "M-G": Mask-Guided Flow.

C. Ablation Study

Table II shows all ablation studies based Res2Net-50 backbone.

1) Effects of MEF Module with/without FE Component: To assess the effectiveness of the MEF module and its FE component, we conduct two ablation studies by comparing the MEF module with/without the FE component to the baseline. The results are shown in 'No.1' and 'No.2' in Table II, which demonstrate that incorporating MEF modules with/without the FE component yields better performance than the baseline. Moreover, the MEF module with the FE component exhibits improved performance, indicating that our designed FE component significantly enhances feature representations.

2) Effects of MECSA Module: MECSA is the core module of MECS-Net. To validate its effectiveness, we added the module to the baseline, and the experimental results are displayed in 'No.3' of Table II. The results show that MECSA improves the baseline performance.

3) Effects of E-G and M-G Components: The effectiveness of MECSA is established. However, the influence of edge and mask cues on the performance of MECSA remains to be explored. To this end, we conduct two sets of ablation experiments wherein we remove the 'E-G' and 'M-G' components, respectively, from the entire MECS-Net. The results are presented in 'No.4' and 'No.5' of Table II, which reveals that the removal of both components adversely impacts the performance of MECS-Net, with 'M-G' playing a more prominent role. These results highlight the critical guiding role played by both components for COD.

IV. CONCLUSION

This letter proposes a mask-and-edge co-guided separable network, namely MECS-Net, for COD. MECS-Net includes two modules, MECSA and MEF, designed to enhance detection performance. Specifically, MECSA incorporates mask and edge cues to learn foreground, background, and edge semantics separately, boosting discriminative feature learning. Our extensive evaluations demonstrate that MECS-Net achieves promising performance.

REFERENCES

[1] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2777–2787.

[2] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, "Concealed object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, Oct. 2022.

[3] H. Bi, C. Zhang, K. Wang, J. Tong, and F. Zheng, "Rethinking camouflaged object detection: Models and datasets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5708–5724, Sep. 2022.

[4] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Comput. Commun.*, vol. 156, pp. 1–10, 2020.

[5] D.-P. Fan et al., "Inf-Net: Automatic COVID-19 lung infection segmentation from CT images," *IEEE Trans. Med. Imag.*, vol. 39, no. 8, pp. 2626–2637, Aug. 2020.

[6] D.-P. Fan et al., "PraNet: Parallel reverse attention network for polyp segmentation," in *Proc. 23rd Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2020, pp. 263–273.

[7] L. Liu et al., "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020.

[8] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, and M.-M. Cheng, "EGNet: Edge guidance network for salient object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8779–8788.

[9] S. K. Singh, C. A. Dhawale, and S. Misra, "Survey of object detection methods in camouflaged image," *IERI Procedia*, vol. 4, pp. 351–357, 2013.

[10] N. U. Bhajantri and P. Nagabhushan, "Camouflage defect identification: A novel approach," in *Proc. IEEE 9th Int. Conf. Inf. Technol.*, 2006, pp. 145–148.

[11] T.-N. Le, T. V. Nguyen, Z. Nie, M.-T. Tran, and A. Sugimoto, "Anabranch network for camouflaged object segmentation," *Comput. Vis. Image Understanding*, vol. 184, pp. 45–56, 2019.

[12] K. Wang, H. Bi, Y. Zhang, C. Zhang, Z. Liu, and S. Zheng, "D^{2}C-Net: A dual-branch, dual-guidance and cross-refine network for camouflaged object detection," *IEEE Trans. Ind. Electron.*, vol. 69, no. 5, pp. 5364–5374, May 2022.

[13] G.-P. Ji, L. Zhu, M. Zhuge, and K. Fu, "Fast camouflaged object detection via edge-based reversible re-calibration network," *Pattern Recognit.*, vol. 123, 2022, Art. no. 108414.

[14] M. Stevens and S. Merilaita, "Animal camouflage: Current issues and new perspectives," *Philos. Trans. Roy. Soc. B: Biol. Sci.*, vol. 364, no. 1516, pp. 423–427, 2009.

[15] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8772–8781.

[16] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, "Context-aware cross-level fusion network for camouflaged object detection," in *Proc. 38th Int. Joint Conf. Artif. Intell.*, Virtual Event / Montreal, Canada, Aug. 19-27, 2021, pp. 1025–1031.

[17] G. Chen, S.-J. Liu, Y.-J. Sun, G.-P. Ji, Y.-F. Wu, and T. Zhou, "Camouflaged object detection via context-aware cross-level fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6981–6993, Oct. 2022.

[18] T. Zhou, Y. Zhou, C. Gong, J. Yang, and Y. Zhang, "Feature aggregation and propagation network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 7036–7047, 2022.

[19] Q. Zhai et al., "MGL: Mutual graph learning for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 32, pp. 1897–1910, 2023.

[20] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10071–10081.

[21] Y. Lv et al., "Simultaneously localize, segment and rank the camouflaged objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11591–11601.

[22] H. Zhu et al., "I can find you! Boundary-guided separated attention network for camouflaged object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 3608–3616.

[23] Y. Sun, S. Wang, C. Chen, and T. Xiang, "Boundary-guided camouflaged object detection," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Vienna, Austria, Jul. 23-29, 2022, pp. 1335–1341.

[24] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, "Detecting camouflaged object in frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4504–4513.

[25] J. Lin, X. Tan, K. Xu, L. Ma, and R. W. Lau, "Frequency-aware camouflaged object detection," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 19, pp. 1–16, 2023.

[26] W. Zhai, Y. Cao, H. Xie, and Z.-J. Zha, "Deep texton-coherence network for camouflaged object detection," *IEEE Trans. Multimedia*, early access, Jul. 04, 2022, doi: 10.1109/TMM.2022.3188401.

[27] P. Li, X. Yan, H. Zhu, M. Wei, X.-P. Zhang, and J. Qin, "FindNet: Can you find me? Boundary-and-texture enhancement network for camouflaged object detection," *IEEE Trans. Image Process.*, vol. 31, pp. 6396–6411, 2022.

[28] J. Zhu, X. Zhang, S. Zhang, and J. Liu, "Inferring camouflaged objects by texture-aware interactive guidance network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3599–3607.

[29] Z. Wu, L. Su, and Q. Huang, "Cascaded partial decoder for fast and accurate salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3907–3916.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[31] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[32] P. Skurowski, H. Abdulameer, J. Błaszczyk, T. Depta, A. Kornacki, and P. Kozieł, "Animal camouflage analysis: Chameleon database," *Unpublished Manuscript*, vol. 2, no. 6, 2018, Art. no. 7.

[33] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4548–4557.

[34] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 248–255.

[35] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 733–740.

[36] D.-P. Fan, C. Gong, Y. Cao, B. Ren, M.-M. Cheng, and A. Borji, "Enhanced-alignment measure for binary foreground map evaluation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 698–704.

[37] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 652–662, Feb. 2021.

[38] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[39] Z. Liu et al., "Swin transformer v2: Scaling up capacity and resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12009–12019.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7-9, 2015.

[41] F. Yang et al., "Uncertainty-guided transformer reasoning for camouflaged object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4146–4155.

[42] Y. Lv, J. Zhang, Y. Dai, A. Li, N. Barnes, and D.-P. Fan, "Towards deeper understanding of camouflaged object detection," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jan. 05, 2023, doi: 10.1109/TCSVT.2023.3234578.