TINYCOD: TINY AND EFFECTIVE MODEL FOR CAMOUFLAGED OBJECT DETECTION

Haozhe Xing[1], *Shuyong Gao*[2], *Hao Tang*[3], *Tsui Qin Mok*[2], *Yanlan Kang*[1], *Wenqiang Zhang*[1,2]

[1]Academy for Engineering & Technology, Fudan University, Shanghai, China
[2]School of Computer Science, Fudan University, Shanghai, China
[3]Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland

ABSTRACT

This paper introduces an effective and tiny model for real-time Camouflaged Object Detection (COD) named **TinyCOD**. It achieves high performance with very low costs ($parameters < 5M$, $flops < 3G$), which can be applied on mobile devices. Specifically, we introduce a simple but effective Adjacent Scale Features Fusion module (ASFF), which can significantly enhance the representation ability of features from a lightweight backbone. Besides, as the edge areas of the camouflaged object often blend into the background, we carefully design an Edge Area Focus module (EAF) to solve this problem. Experimental results on COD datasets prove that the proposed method achieves state-of-the-art performance compared with other methods. Source codes will be available.

Index Terms— Camouflaged object detection, lightweight model, feature fusion, edge features, image segmentation.

1. INTRODUCTION

The goal of Camouflaged Object Detection (COD) [1] is to find and segment the target hidden in the environment, which is just the opposite of Salient Object Detection (SOD) [2]. Generally, the color, texture, and shape of the camouflaged object are very similar to the surrounding environment, which makes COD become a more difficult task than SOD. COD has many applications, such as agriculture, medical care [3], and marine animal segmentation [4]. Due to its great application value, more and more researchers are working on COD tasks.

The difficulty of the COD task lies in finding the camouflaged object and completely segmenting the edges of camouflaged objects. Researchers attempt to solve these problems with deep learning techniques and achieve excellent performance. Imitating the predatory process of animals in nature, SINet [1, 6], PFNet [7] design their methods based on the search-recognize architecture. SINet introduces a large COD dataset, named COD10K. PFNet proposes a focus module, which uses interference minimization strategies to refine the prediction. Zhai et al. [8] put forward a mutual learning approach for COD and camouflaged object-aware edge extraction tasks based on the graph structure. Lv et al. [9] introduce the concept of camouflage level, they use RankNet to rank the camouflage objects and provide a new test dataset for COD tasks, named NC4K. Pang et al. [10] proposed the mixed triple inputs network obtaining a good performance but costs highly. C2FNet [11, 12] adopts the cross-level feature fusion strategy to improve its performance. Although these methods perform well on COD, they are resource-starved in terms of calculated costs.

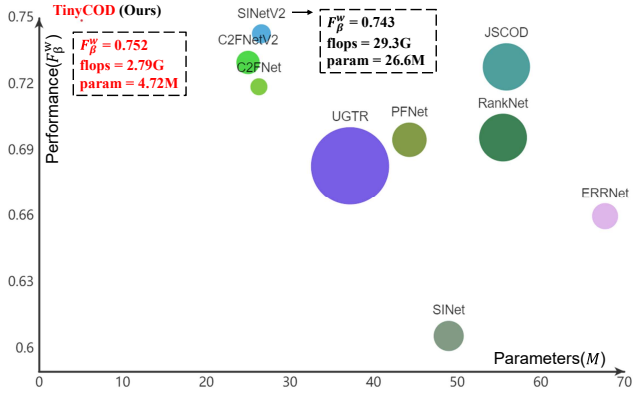

Fig. 1: The performance on CAMO [5] datasets, parameters, and flops compared with other SOTA models. The area of the circle represents the costs of flops (**ours close to a point**). Our model achieves SOTA performance with very low costs.

Our goal is to design a lightweight COD model that can be used on mobile devices. Different from previous works, as shown in Figure 1, our model costs very low parameters and flops with performance guaranteed. To the best of our knowledge, this is the first tiny model for camouflaged object detection. We purpose a lightweight COD model based on a lightweight backbone, but the representation capability of features from the lightweight backbone may not be strong enough for COD tasks. Thus, we develop an adjacent scale features fusion module that can improve the model perfor-

This work was supported by National Key RD Program of China (2020AAA0108301), National Natural Science Foundation of China (No.62072112), Scientific and Technological innovation action plan of Shanghai Science and Technology Committee (No.22511102202), Fudan University-CIOMP Joint Fund (FC2019-005), Double First-class Construction Fund (No. XM03211178).

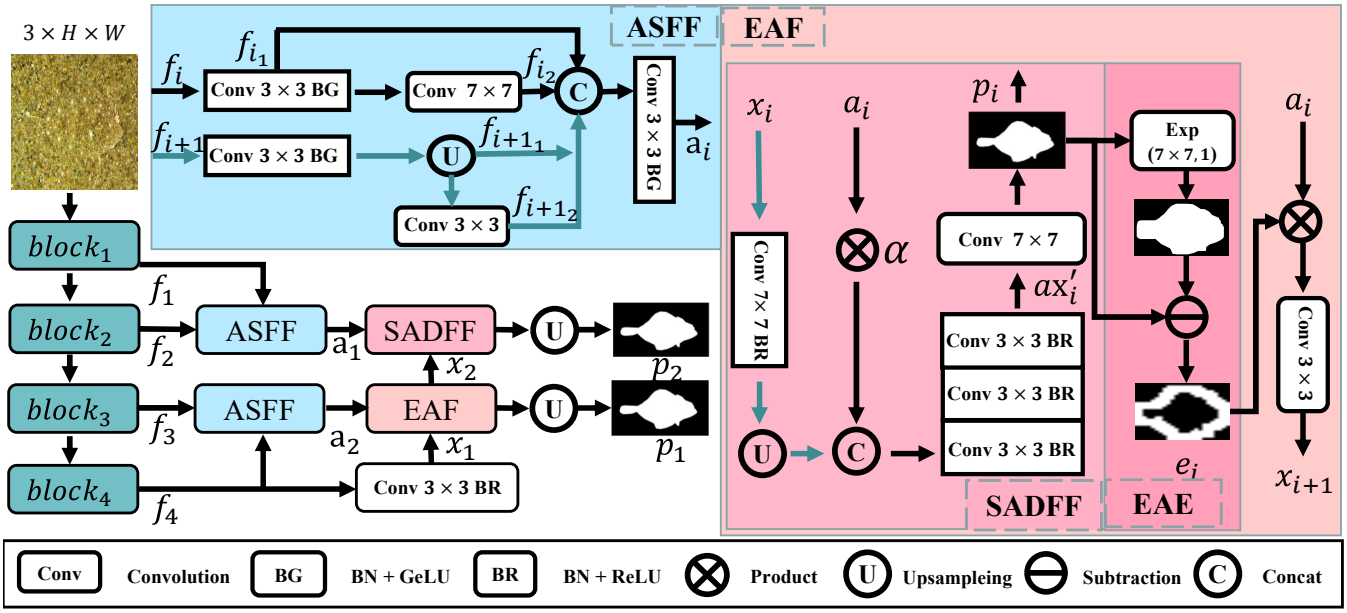

Fig. 2: The architecture of TinyCOD. Our model contains two key modules: the ASFF module and the EAF module. The EAF module includes the SADFF module (also used separately in the pipeline) and the EAE module.

mance with low calculation consumption. In addition, it is more difficult to segment the edge area of camouflaged objects precisely. To solve this problem, we propose a module named the edge area focus module. Different from [8, 13], the edge area focus module can make the model focus more on the edge areas without using edges as supervision.

Our contributions can be summarized as follows:

- We propose a tiny ($parameters < 5M$, $flops < 3G$) and effective model named TinyCOD for camouflaged object detection, which has great application value on mobile devices.
- We carefully design a simple but effective adjacent scale features fusion module to enhance feature representation and a novel edge area focus module to make the model pay more attention to the edge areas without using edges as supervision, both of them can improve the model performance at a very low cost.
- Experiments have demonstrated that our method achieves state-of-the-art performance at a very low cost compared with other state-of-the-art COD methods.

2. THE PROPOSED METHOD

The architecture of our method is shown in Figure 2. We utilize TinyNet (TinyNet-a) [14] as our lightweight backbone and use the last 4 different layers $block_i (i = 1, 2, 3, 4)$ of it to extract features $f_i (i = 1, 2, 3, 4)$ from images. To improve the performance with a lower cost, we introduce two simple yet effective modules: the Adjacent Scale Features Fusion module (ASFF) and the Edge Area Focus module (EAF).

2.1. Adjacent Scale Features Fusion Module

Due to the great difficulty of the COD task, a lightweight backbone may not extract rich feature information. Since adjacent scale features have high similarity feature information, fusing adjacent features can make the model obtain better-integrated features than other feature fusion strategies at a lower cost. Thus, we utilize the ASFF module to fusion adjacent scale features to enhance the representation ability of features. As shown in Figure 2, the ASFF module consists of five convolution blocks with two adjacent scale features $f_i \in \mathbb{R}^{c_i \times h_i \times w_i}$ and f_{i+1} as inputs. Firstly, both f_i and f_{i+1} are fed into a block 'Conv 3 × 3 BG' which consists of a 3 × 3 convolution block, a batch normalization layer and GeLU activation function to get features $f_{i_1} \in \mathbb{R}^{\frac{1}{2}c_{out} \times h_i \times w_i}$ and $f_{i+1_1} \in \mathbb{R}^{\frac{1}{2}c_{out} \times h_{i+1} \times w_{i+1}}$. Then we utilize up-sampling operation on f_{i+1_1} to make it share the same size with f_{i_1}. Thirdly, $f_{i_2} \in \mathbb{R}^{1 \times h_i \times w_i}$ are obtained by inputting f_{i_1} into a 7 × 7 convolution block. We also input f_{i+1_1} into a 3 × 3 convolution block to get $f_{i+1_2} \in \mathbb{R}^{1 \times h_i \times w_i}$. Finally, features are fused to get $a_i \in \mathbb{R}^{c_{out} \times h_i \times w_i}$. The process of features fusion can be formulated as follows:

$$a_i = C3BG(Cat(f_{i_1}, f_{i+1_1}, f_{i_2}, f_{i+1_2})), \quad (1)$$

where $C3BG$ is a block including a convolution block with 3 × 3 kernel size, a batch normalization layer, and GeLU activation function; Cat refers to concatenate operation.

2.2. Edge Area Focus Module

The edge areas of camouflaged objects often blend into the environment as Img shown in Figure 3, which leads to many

Table 1: Comparison of our proposed method with other COD state-of-the-art methods on four benchmarks. All the results are evaluated with the same evaluation codes. Bold indicates our methods.

Method	Source	Param	Flops	CAMO				CHAMELEON				COD10K				NC4K			
				$M\downarrow$	$E_\phi^{ad}\uparrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$E_\phi^{ad}\uparrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$E_\phi^{ad}\uparrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$	$M\downarrow$	$E_\phi^{ad}\uparrow$	$S_\alpha\uparrow$	$F_\beta^w\uparrow$
SINet [1]	CVPR'20	49.0M	46.5G	0.100	0.835	0.752	0.606	0.044	0.899	0.869	0.740	0.051	0.797	0.771	0.551	0.058	0.883	0.808	0.723
TINet [13]	AAAI'21	-	-	0.087	0.847	0.781	0.678	-	-	-	-	0.043	0.848	0.793	0.635	0.055	0.882	0.829	0.734
TANet [15]	TCSVT'21	-	-	0.083	0.834	0.793	0.690	0.036	0.911	0.888	0.786	0.041	0.848	0.803	0.629	-	-	-	-
MGLS [8]	CVPR'21	73.7M	757.2G	0.090	0.850	0.772	0.664	0.032	0.920	0.892	0.802	0.037	0.850	0.811	0.654	0.055	0.884	0.829	0.731
PFNet [7]	CVPR'21	44.3M	53.2G	0.085	0.855	0.782	0.695	0.033	0.942	0.882	0.810	0.040	0.868	0.800	0.660	0.053	0.894	0.829	0.745
RankNet [9]	CVPR'21	55.5M	73.8G	0.080	0.859	0.787	0.696	0.031	0.936	0.890	0.822	0.037	0.883	0.804	0.673	0.048	0.904	0.840	0.766
C2FNet [11]	IJCAI'21	26.3M	26.3G	0.080	0.865	0.796	0.719	0.032	0.932	0.888	0.828	0.036	0.886	0.813	0.686	0.049	0.901	0.838	0.762
JSCOD [16]	CVPR'21	55.9M	73.1G	0.073	0.872	0.800	0.728	0.030	0.943	0.891	0.833	0.035	0.882	0.809	0.684	0.047	0.906	0.842	0.771
UGTR [17]	ICCV'21	37.2M	242.7G	0.864	0.858	0.783	0.683	0.031	0.920	0.888	0.794	0.036	0.850	0.817	0.665	0.052	0.888	0.839	0.746
SINetV2 [6]	TPAMI'21	26.6M	29.3G	0.071	0.884	0.820	0.743	0.030	0.930	0.888	0.816	0.037	0.864	0.815	0.680	0.048	0.901	0.847	0.770
ERRNet [18]	PR'22	67.7M	40.2G	0.088	0.831	0.761	0.660	0.036	0.923	0.877	0.805	0.044	0.856	0.780	0.629	-	-	-	-
C2FNetV2 [12]	TCSVT'22	25.2M	36.1G	0.077	0.867	0.799	0.730	0.028	0.947	0.893	0.845	0.036	0.891	0.811	0.691	-	-	-	-
TinyCOD	Ours	**4.72M**	**2.79G**	**0.066**	**0.890**	**0.822**	**0.752**	**0.030**	**0.931**	**0.887**	**0.814**	**0.036**	**0.877**	**0.811**	**0.678**	**0.047**	**0.903**	**0.843**	**0.766**

methods missing some edge area information. To solve this problem, we proposed the Edge Area Focus (EAF) module, which contains the Shallow And Deep Features Fusion (SADFF) module and Edge Area Expand (EAE) module. The SADFF module aims to rich the feature representation and the EAE module can make the model pay more attention to the edge areas without utilizing edges as supervision. More importantly, the EAF module costs little to nothing but can bring significant performance improvement.

As shown in Figure 2, the SADFF module takes deep features x_i and shallow features a_i as inputs. Deep features x_i after a 7×7 convolution block, a batch normalization layer, ReLU activation function, and up-sampling operation share the same shape with a_i. Then we concat $x_i^{'}$ and $\alpha \times a_i$ to get ax_i, where α is a learnable parameter. $ax_i^{'}$ is obtained by making ax_i pass through three continuous 'Conv 3×3 BN' blocks. Finally, we input $ax_i^{'}$ into a 7×7 convolution block to get output p_i of the SADFF module. The EAE module takes p_i as input. We utilize sigmoid function and expand operation (Exp) on p_i to get expanded prediction $p_i^{'}$, and then subtract p_i from $p_i^{'}$ to get the output e_i. Finally, $a_i \times e_i$ is sent into 3×3 convolution block to get the output x_{i+1} of the EAF module. The process of obtaining x_{i+1} can be formulated as follows:

$$\begin{aligned} e_i &= Exp(S(p_i)) - S(p_i), \\ x_{i+1} &= Conv3(a_i \times e_i), \end{aligned} \tag{2}$$

where Exp denotes the morphological dilation operation with 7×7 kernel size and one iteration; S represents sigmoid function; $Conv3$ is a 3×3 convolution block.

2.3. Loss Function

There are two predictions $p_i (i = 1, 2)$ of our model. All predictions are resized to the same resolution with input. Following other methods [7, 11, 12], our basic loss function $L_{basic-loss}$ [19] combines with the weighted intersection-over-union loss L_{Iou}^w and the weighted binary cross-entropy loss L_{bce}^w. The overall loss function can be described as:

$$\begin{aligned} L_{basic-loss}^i &= L_{bce}^w(p_i, GT) + L_{Iou}^w(p_i, GT), \\ L_{overall-loss} &= L_{basic-loss}^1 + \alpha L_{basic-loss}^2, \end{aligned} \tag{3}$$

where p_i represents the i-th prediction, GT is the ground-truth; $\alpha = 2$; L_{basic}^2 is calculated by the final prediction.

3. EXPERIMENTS AND RESULTS

3.1. Experiments

Datasets. Following most COD methods, we train our model on CAMO [5] and COD10K [6], test it on CHAMELEON [20], CAMO, COD10K, and NC4K [9]. CAMO contains 1,250 camouflaged object images, 1,000 images for training, and 250 for testing. CHAMELEON includes 76 camouflaged objects images for testing. COD10K contains 3,040 camouflaged objects images for training and 2,026 for testing. NC4K is a testing dataset including 4,121 images.

Evaluation Metrics. We use four evaluation metrics that are widely used in COD tasks, including Structure-measure (S_α) [21], Adaptive E-measure (E_ϕ^{ad}) [22], Weighted F-measure (F_β^w) [23], Mean Absolute Error (M). E_ϕ^{ad} evaluates local and global information. S_α is employed to calculate the structural similarity of objects and regions. F_β^w is utilized for weighted precision and recall. M is to calculate the pixel-wise difference between the prediction and the ground-truth.

Implementation Details. The training process used 4,040 camouflage object images from CAMO and COD10K. We resize the input images into a resolution of 384×384 and augment the input images by randomly flipping, cropping, rotation, and color enhancement. The optimizer is Adam optimizer. We use the CosineAnnealingLR strategy to adjust the

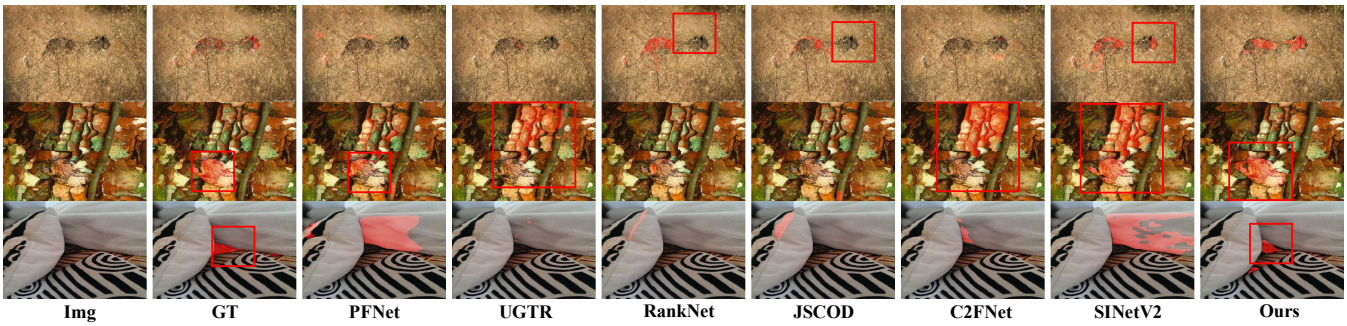

Fig. 3: Visualization comparison of our method compared with six state-of-the-art COD methods. Predictions of our method are obviously better than others.

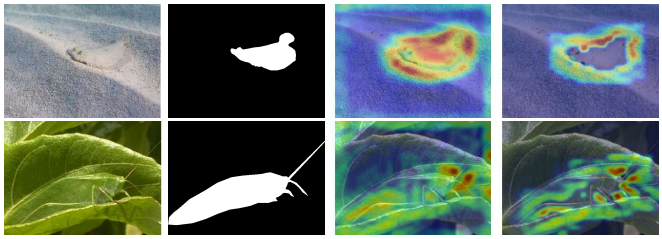

Fig. 4: Visualization of features in the EAF module.

learning rate. In addition, the batch size is 16, the training epoch is set to 100, and the basic learning rate is 0.001.

3.2. Compared with the State-of-the-Art Methods

We compared our model with twelve existing state-of-the-art methods including SINet [1], PFNet [7], MGL [8], ERRNet [18], JSCOD [16], RankNet [9], C2FNet [11], TANet [15], SINetV2 [6], UGTR [17], TINet [13], and C2FNetV2 [12] on four benchmarks. To ensure a fair comparison, we downloaded the results map of these methods and evaluated them with the same evaluation tools. For quantitative evaluation, it can be seen in Table 1 that our model achieves better performance compared with JSCOD but only using **3.8% flops**, **8.5% parameters** of it. For qualitative evaluation shown in Figure 3, compared with other SOTA methods, our model can identify objects under various camouflages, such as background matching, small, and occluded objects.

3.3. Ablation Study

We conducted ablation studies to validate the effectiveness of our proposed modules. As shown in Table 2, Base means the baseline model. We use basic convolution and feature fusion methods to replace the ASFF module, SADFF module, and EAF module in TinyCOD (as shown in Figure 2) to serve as the baseline model. ASFF is the adjacent scale features fusion module; EAF represents the edge area focus module. MobileV2-COD is our model used MobileNetV2 [24] as the backbone. Tiny-SINetV2 represents SINetV2 utilizes the same backbone (TinyNet) with our method.

Table 2: Ablation study of the proposed method.

Method	Param	Flops	COD10K			
			$M \downarrow$	$E_\phi^{ad} \uparrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$
Tiny-SINetV2	4.96M	4.24G	0.044	0.842	0.781	0.619
MobileV2-COD	4.50M	4.15G	0.039	0.863	0.799	0.654
(a) Base	4.63M	2.69G	0.049	0.806	0.780	0.598
(b) Base + SADFF	4.65M	2.75G	0.043	0.839	0.791	0.632
(c) Base + ASFF	4.69M	2.72G	0.038	0.855	0.807	0.659
(d) Base + EAF	4.66M	2.76G	0.037	0.867	0.808	0.670
(e) Ours	**4.72M**	**2.79G**	**0.036**	**0.877**	**0.811**	**0.678**

In Table 2, compared with Tiny-SINetV2 (row 1), F_β^w of our method increases by 5.9% while flops decrease by 1.45G. From row (2) and row (e), we can find that our method achieves excellent performance under both lightweight backbones (MobileNetV2-120d, TinyNet-a). F_β^w of the model with ASFF (row (c)) compared with the model without ASFF (row (a)) on COD10K increases by 6.1%. Compared with row (a) and (d), F_β^w with EAF increases by 7.2 % on COD10K. Both ASFF and EAF bring huge performance gains with very low costs. As features x_{i+1} shown in Figure 4, EAF does make the model pay more attention to the edge area, which proves the effectiveness of the EAF module.

4. CONCLUSION

We proposed a tiny but effective model (achieves state-of-the-art performance) named TinyCOD, which is the first low costs ($parameters < 5M$) network that has great application value and can be utilized on mobile devices for camouflaged object detection. In the TinyCOD, we introduce two tiny but effective modules called the Adjacent Scale Features Fusion module (ASFF) and Edge Area Focus module (EAF). The ASFF module can rich feature information from the lightweight backbone, and the EAF module can make the model focus more on the edge area of the camouflaged object, which can greatly improve the performance of the model.

5. REFERENCES

[1] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao, "Camouflaged object detection," in *CVPR*, 2020, pp. 2777–2787.

[2] Shuyong Gao, Qianyu Guo, Wei Zhang, Wenqiang Zhang, and Zhongwei Ji, "Dual-stream network based on global guidance for salient object detection," in *ICASSP*. IEEE, 2021, pp. 1495–1499.

[3] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," in *ICASSP*. IEEE, 2020, pp. 1055–1059.

[4] Ruizhe Chen, Zhenqi Fu, Yue Huang, En Cheng, and Xinghao Ding, "A robust object segmentation network for under water scenes," in *ICASSP*. IEEE, 2022, pp. 2629–2633.

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

[6] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao, "Concealed object detection," *TPAMI*, 2021.

[7] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan, "Camouflaged object segmentation with distraction mining," in *CVPR*, 2021, pp. 8772–8781.

[8] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan, "Mutual graph learning for camouflaged object detection," in *CVPR*, 2021, pp. 12997–13007.

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

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

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

[12] Geng Chen, Si-Jie Liu, Yu-Jia Sun, Ge-Peng Ji, Ya-Feng Wu, and Tao Zhou, "Camouflaged object detection via context-aware cross-level fusion," *TCSVT*, 2022.

[13] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu, "Inferring camouflaged objects by texture-aware interactive guidance network," in *AAAI*, 2021, vol. 35, pp. 3599–3607.

[14] Kai Han, Yunhe Wang, Qiulin Zhang, Wei Zhang, Chunjing Xu, and Tong Zhang, "Model rubik's cube: Twisting resolution, depth and width for tinynets," *NIPS*, vol. 33, pp. 19353–19364, 2020.

[15] Jingjing Ren, Xiaowei Hu, Lei Zhu, Xuemiao Xu, Yangyang Xu, Weiming Wang, Zijun Deng, and Pheng-Ann Heng, "Deep texture-aware features for camouflaged object detection," *TCSVT*, 2021.

[16] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai, "Uncertainty-aware joint salient object and camouflaged object detection," in *CVPR*, 2021, pp. 10071–10081.

[17] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Ao Luo, Hong Cheng, and Deng-Ping Fan, "Uncertainty-guided transformer reasoning for camouflaged object detection," in *CVPR*, 2021, pp. 4146–4155.

[18] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu, "Fast camouflaged object detection via edge-based reversible re-calibration network," *PR*, vol. 123, pp. 108414, 2022.

[19] Jun Wei, Shuhui Wang, and Qingming Huang, "F^3net: fusion, feedback and focus for salient object detection," in *AAAI*, 2020, vol. 34, pp. 12321–12328.

[20] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł, "Animal camouflage analysis: Chameleon database," *Unpublished manuscript*, vol. 2, no. 6, pp. 7, 2018.

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

[22] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji, "Enhanced-alignment measure for binary foreground map evaluation," *arXiv preprint arXiv:1805.10421*, 2018.

[23] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal, "How to evaluate foreground maps?," in *CVPR*, 2014, pp. 248–255.

[24] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.