Rethinking Detecting Salient and Camouflaged Objects in Unconstrained Scenes
⟨ Supplementary Material ⟩

Zhangjun Zhou[1*] Yiping Li[1*] Chunlin Zhong[1*] Jianuo Huang[1]
Jialun Pei[2] Hua Li[3] He Tang[1†]

[1]School of Software Engineering, Huazhong University of Science and Technology
[2]School of Computer Science and Engineering, The Chinese University of Hong Kong
[3] School of Computer Science and Technology, Hainan University

Appendix

We summarize the supplementary material from the following aspects:

Table of contents:

1. Results of more models in Table 1 & 4.

We selected five representative models to highlight the misinterpretation in the manuscript. And we conducted experiments with more models to demonstrate that this is a widespread issue, as shown below. The misinterpretation is largely mitigated after training on the USC12K dataset.

Table 1. Misinterpretation is prevalent when SOD/COD models applied across tasks. **Left:** Inference results of SOD models on COD datasets. **Right:** Inference results of COD models on SOD datasets. The metric is F_β^ω. EV refers to the Expected Value. After training with USC12K, misinterpretation is largely eliminated.

	SOD Models	COD Datasets		EV
		COD10K	NC4K	
Before	GateNet	0.4369	0.6036	0
	MDSAM	0.6205	0.7303	0
	Spider	0.7590	0.8344	0
After	GateNet	0.0059	0.0493	0
	MDSAM	0.0099	0.1200	0
	Spider	0.0089	0.0594	0

	COD Models	SOD Datasets		EV
		DUTS	HKU-IS	
Before	ZoomNet	0.7008	0.6790	0
	CamoFormer	0.7921	0.7836	0
	Spider	0.8621	0.9089	0
After	ZoomNet	0.0777	0.0599	0
	CamoFormer	0.1192	0.0764	0
	Spider	0.0647	0.0486	0

2. CSCS Metric

Contrary to the Intersection over Union (IoU) that measures accuracy for a single class, the Camouflage-Saliency Confusion Score (CSCS) assesses the misclassification between two distinct classes. The CSCS, designed to evaluate the confusion between camouflaged and salient objects, is calculated as follows:

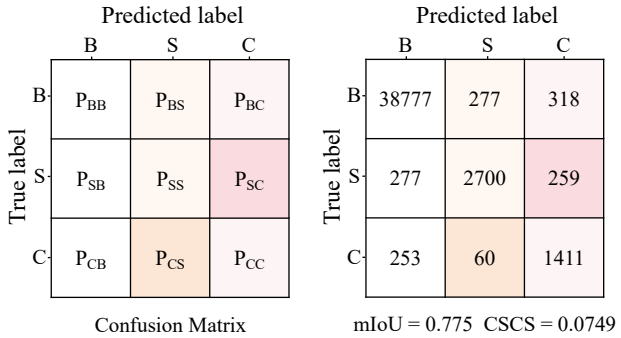
$$\text{CSCS} = \frac{1}{2}\left(\frac{\mathcal{P}_{CS}}{\mathcal{P}_{BS}+\mathcal{P}_{SS}+\mathcal{P}_{CS}} + \frac{\mathcal{P}_{SC}}{\mathcal{P}_{BC}+\mathcal{P}_{SC}+\mathcal{P}_{CC}}\right), \quad (1)$$

where $\mathbb{P} = \{\mathcal{P}_{\lambda\theta} \,|\, \lambda \in \Theta, \theta \in \Theta\}$, $\Theta = \{B, C, S\}$, the B, C and S denote background, camouflage and saliency. A lower CSCS value indicates a stronger ability of the network to discriminate between salient and camouflaged objects. $\mathcal{P}_{CS}$ represents the label as camouflage but is predicted as saliency. We aim to minimize the misclassification of camouflaged pixels as salient, ensuring the network correctly distinguishes between camouflaged and salient objects. The same applies to $\mathcal{P}_{SC}$. As shown in Figure 2, we present the confusion matrix of the proposed USCNet on the USC12K test set. Our model balances improvements across all metrics, achieving a mIoU of 0.775 and a CSCS of 0.0749.

Image GT Prediction

Figure 1. The illustration of $\mathcal{P}_{BS}$, $\mathcal{P}_{SS}$, $\mathcal{P}_{CS}$, $\mathcal{P}_{BC}$, $\mathcal{P}_{SC}$, and $\mathcal{P}_{CC}$ in the CSCS metric. The **red** mask represents the salient regions, and the **green** mask denotes the camouflaged regions.

*Both authors contributed equally to this research.
†Corresponding author: He Tang (hetang@hust.edu.cn).



(a) Illustration (b) USCNet

Figure 2. Confusion matrix of our USCNet on the USC12K test set. The units of the values in the confusion matrix are in tens of thousands (**1E+04**).

3. Performance in Detecting Objects of Different Sizes

To evaluate the model's ability to detect objects of varying sizes, we employ several metrics: AUC ↑, SI-AUC ↑, F_m^β ↑, SI-F_m^β ↑, $F_{\max}^\beta$ ↑, SI-$F_{\max}^\beta$ ↑, E_m ↑. From Table 2 and Table 3, it can be observed that, compared to the size-sensitive(e.g., AUC ↑ and F_m^β ↑) and size-invariance metrics(e.g., SI-AUC ↑ and SI-F_m^β ↑), our method exhibits smaller performance fluctuations, demonstrating its robustness to variations in object size and number in the scene.

4. Results on Popular COD and SOD Datasets

To further validate the effectiveness and robustness of our method regarding generalizability, we conduct tests on popular SOD datasets (DUTS [32], HKU-IS [17], and DUT-OMRON [39]) and COD datasets (CAMO [16], COD10K [7], and NC4K [24]), with all methods uniformly trained using our USC12K dataset. We adopt five metrics that are widely used in COD and SOD tasks [8, 34]. These metrics include maximal F-measure ($F_\beta^{\max}$ ↑) [1], weighted F-measure (F_β^ω ↑) [25], Mean Absolute Error (MAE, M ↓) [28], Structural measure (S-measure, S_α ↑) [5], and mean Enhanced alignment measure (E-measure, E_ϕ^{m} ↑) [6]. As shown in Table 4 and Table 5, our USCNet achieves state-of-the-art performance on these datasets through parameter-efficient fine-tuning. This further confirms the strong capability of our method to accurately identify both salient and camouflaged objects in unconstrained environments. This achievement is attributed to the exceptional versatility of SAM in class-agnostic segmentation tasks and the discriminative ability of our specially designed ARM for distinguishing between salient and camouflaged objects.

5. More Technical Details

All models are retrained using the training set of USC12K with an input image resolution of 352×352. Horizontal flipping and random cropping are applied for data augmentation. The experiments are conducted in PyTorch on one NVIDIA L40 GPU. For our model, we use the hiera-large version of SAM2 following the SAM2-Adapter [3]. AdamW optimizer is used with a warm-up strategy and linear decay strategy. The initial learning rate is set to 0.0001. The batch size is set to 24, and the maximum number of epochs is set to 90.

Backbone of models. The models compared can be divided into two categories based on their papers: one is full-tuning models, and the other is parameter-efficient fine-tuning (PEFT) models. (i)Full Tuning models: Include all SOD and COD methods and VSCode in the Unified Method. For fairness, the models compared are all trained according to the configurations specified in their original papers. (ii)PEFT models: SAM-Adapter, SAM2-Adapter, EVP in the Unified Method and our model. The backbone architectures across various models consist of several types. For full tuning, VST employs a transformer encoder based on T2T-ViT [41], while SINet-V2 utilizes Res2Net-50 [9]. VSCode uses Swin-T [21], and ICEG adopts Swin-B [21]. PRNet is based on the SMT backbone [18], and both CamoDiffusion, CamoFormer, and PGT use PVTv2-b4 [35]. Other models generally rely on ResNet-50 [12] with pre-trained weights from ImageNet [4]. In the case of PEFT models, EVP uses SegFormer-B4 [38] as its base, SAM-Adapter uses the default ViT-H version of SAM [15], and both SAM2-Adapter and our model employ the hiera-large version of SAM2 [30].

Training and Inference. For traditional SOD and COD models: USC12K is defined by three aspects: saliency, camouflage, and background. Conventional methods for COD and SOD are crafted for dichotomous mapping tasks and don't seamlessly transition to the nuanced demands of the USC12K benchmark. Inspired by seminal works in semantic segmentations [22, 31], we retool the output layers of our models to yield a tripartite representation for saliency, camouflage, and background. This is achieved by harnessing a softmax layer to generate a predictive mapping. We employ a cross-entropy loss function to refine the model, which is congruent with our overarching methodological framework. For unified models: VSCode and EVP, which require task-specific prompts for each dataset, we create two copies of the USC12K training set. One copy is used for SOD, with the ground truth being the SOD-only mask, and is used to train the prompts corresponding to the SOD task. The other copy is used for COD, with the ground truth being the COD-only mask, and is used to train the prompts corresponding to the COD task. VSCode is trained once using all 16,800 images (two copies of 8,400 images), while EVP is

Table 2. Performance of different models detecting salient objects on USC12K testing set.

Task	Model	Update Params(M)	USC12K-SOD						
			AUC ↑	SI-AUC↑	F_m^β↑	SI-F_m^β↑	$F_{\max}^\beta$↑	SI-$F_{\max}^\beta$↑	E_m↑
SOD	GateNet [43]	128	.810	.812	.696	.754	.706	.764	.775
	F3Net [36]	26	.828	.826	.722	.765	.734	.777	.803
	MSFNet [42]	28	.832	.831	.726	.772	.735	.782	.805
	VST [19]	43	.777	.777	.642	.732	.650	.741	.742
	EDN [37]	43	.831	.830	.726	.769	.736	.780	.804
	ICON [44]	32	.821	.832	.702	.764	.711	.774	.795
COD	SINetV2 [8]	27	.843	.842	.755	.783	.765	.793	.827
	PFNet [26]	47	.820	.822	.712	.756	.724	.767	.799
	ZoomNet [27]	33	.821	.823	.710	.765	.720	.774	.791
	FEDER [10]	44	.841	.842	.742	.784	.750	.796	.820
	ICEG [11]	100	.830	.825	734	.762	.743	.770	.831
	PRNet [13]	13	.851	.845	.742	.779	.750	.792	.832
	CamoFormer [40]	71	.844	.843	.750	.782	.758	.790	.821
	PGT [33]	68	.831	.828	.717	.773	.727	.784	.791
	SAM2-Adapter [3]	4.36	.847	.847	.741	.783	.751	.794	.816
Unified	VSCode [23]	60	.843	.842	.749	.776	.768	.789	.822
	EVP [20]	4.95	.850	.847	.751	.782	.771	.792	.830
	USCNet(Ours)	**4.04**	**.853**	**.850**	**.761**	**.787**	**.772**	**.798**	**.833**

Table 3. Performance of different models detecting camouflaged objects on USC12K testing set.

Task	Model	Update Params(M)	USC12K-COD						
			AUC↑	SI-AUC↑	F_m^β↑	SI-F_m^β↑	$F_{\max}^\beta$↑	SI-$F_{\max}^\beta$↑	E_m↑
SOD	GateNet [43]	128	.692	.687	.443	.558	.453	.569	.651
	F3Net [36]	26	.695	.687	.449	.564	.458	.574	.649
	MSFNet [42]	28	.698	.691	.455	.565	.465	.576	.659
	VST [19]	43	.626	.625	.303	.536	.312	.546	.524
	EDN [37]	43	.709	.703	.476	.575	.485	.585	.670
	ICON [44]	32	.663	.663	.384	.549	.394	.560	.587
COD	SINetV2 [8]	27	.715	.705	.505	.588	.514	.598	.690
	PFNet [26]	47	.678	.672	.429	.544	.440	.555	.630
	ZoomNet [27]	33	.657	.653	.394	.545	.405	.556	.588
	FEDER [10]	44	.710	.703	.486	.567	.497	.578	.689
	ICEG [11]	100	.730	.717	.525	.601	.532	.609	.719
	PRNet [13]	13	.705	.695	.454	.569	.464	.579	.652
	CamoFormer [40]	71	.756	.745	.565	.626	.575	.636	.743
	PGT [33]	68	.746	.734	.527	.596	.539	.607	.715
	SAM2-Adapter [3]	4.36	.770	.761	.575	.637	.585	.647	.746
Unified	VSCode [23]	60	.735	.727	.519	.601	.525	.597	.722
	EVP [20]	4.95	.695	.684	.485	.577	.494	.587	.650
	USCNet(Ours)	**4.04**	**.801**	**.794**	**.610**	**.658**	**.619**	**.667**	**.795**

trained twice on the two separate training sets (each containing 8,400 images) to obtain the two task-specific prompts. During inference, all unified models perform inference on the testing set of USC12K twice, with the corresponding prompt enabled for each task. The first inference run generates the SOD results, and the second inference run generates the COD results. The final prediction is obtained by merging the SOD and COD predictions. For overlapping pixels, the aspect with the higher prediction value between the two tasks is chosen as the final aspect for that pixel.

6. More USC12K Dataset Detail and Examples

Object category distribution. We obtain an initial coarse classification using CLIP [29], followed by manual verification and refinement. Except for images collected from COD10K [7], which already include camouflage object category labels, all other objects require classification. Then we assign category labels to each image, covering 9 superclasses and 179 sub-classes. Figure 3 illustrates the class breakdown of our USC12K dataset.

Object number distribution. Our USC12K dataset contains images with different numbers of objects. For clarify, we have counted the distribution of images with different

Table 4. Generalization performance of related methods on the DUTS, HKU-IS, and DUT-OMRON test sets. ↑ / ↓ represents the higher/lower the score, the better.

Task	Model	Update Params(M)	DUTS					HKU-IS					DUT-OMRON				
			$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$
SOD	GateNet [43]	128	.666	.644	.062	.755	.765	.804	.785	.049	.841	.857	.634	.603	.079	.747	.751
	F3Net [36]	26	.703	.683	.055	.783	.794	.832	.816	.044	.853	.881	.638	.615	.073	.747	.758
	MSFNet [42]	28	.651	.638	.063	.749	.758	.824	.806	.045	.853	.877	.641	.611	.076	.751	.764
	VST [19]	43	.630	.610	.061	.744	.749	.777	.760	.052	.820	.851	.580	.560	.073	.720	.715
	EDN [37]	43	.692	.676	.053	.784	.785	.820	.806	.043	.852	.873	.616	.597	.071	.742	.735
	ICON [44]	32	.679	.647	.069	.769	.785	.814	.787	.051	.843	.874	.615	.576	.099	.728	.738
COD	SINetV2 [8]	27	.732	.710	.052	.801	.821	.838	.822	.046	.847	.884	.665	.642	.068	.763	.786
	PFNet [26]	47	.691	.668	.060	.775	.790	.818	.801	.048	.843	.876	.643	.614	.075	.747	.764
	ZoomNet [27]	33	.729	.709	.053	.801	.813	.785	.774	.051	.830	.842	.623	.601	.075	.742	.735
	FEDER [10]	44	.736	.714	.052	.808	.821	.839	.827	.045	.869	.881	.645	.615	.077	.755	.760
	PRNet [13]	13	.773	.756	.043	.830	.849	.840	.833	.044	.857	.880	.708	.685	.057	**.796**	.808
	ICEG [11]	100	.719	.700	.050	.789	.820	.832	.815	.045	.848	**.896**	.664	.645	.061	.762	.785
	CamoFormer [40]	71	.733	.715	.049	.813	.819	.838	.817	.046	.857	.884	.687	.661	.066	.783	.793
	PGT [33]	68	.686	.670	.053	.786	.779	.819	.802	.044	.855	.871	.642	.619	.068	.758	.754
	SAM-Adapter [2]	4.11	.761	.746	.048	.834	.796	.822	.806	.043	.836	.869	.708	.685	.059	.793	.802
	SAM2-Adapter [3]	4.36	.776	.762	.041	.831	.848	.831	.828	**.042**	.849	.881	.706	.692	**.056**	.790	.810
Unified	VSCode [23]	60	.724	.706	.060	.795	.812	.834	.830	.043	.851	.885	.636	.608	.075	.748	.753
	EVP [20]	4.95	.769	.750	.045	.833	.836	.835	.832	.043	.852	.878	**.710**	.692	.057	.794	.810
	USCNet(Ours)	**4.04**	**.784**	**.780**	**.040**	**.835**	**.852**	**.844**	**.840**	**.042**	**.860**	.886	**.710**	**.697**	**.056**	**.796**	**.814**

Table 5. Generalization performance of related methods on CAMO, COD10K, and NC4K test set. ↑ / ↓ represents the higher/lower the score, the better.

Task	Model	Update Params(M)	CAMO					NC4K					COD10K				
			$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$	$F_\beta^{\max}\uparrow$	$F_\beta^{\omega}\uparrow$	$M\downarrow$	$S_\alpha\uparrow$	$E_\phi^{\mathrm{m}}\uparrow$
SOD	GateNet [43]	128	.573	.542	.109	.666	.680	.562	.529	.047	.707	.724	.675	.645	.066	.752	.777
	F3Net [36]	26	.538	.506	.117	.643	.657	.576	.539	.047	.712	.744	.661	.633	.070	.738	.773
	MSFNet [42]	28	.568	.535	.113	.661	.682	.543	.534	.052	.692	.719	.671	.645	.067	.747	.778
	VST [19]	43	.484	.455	.109	.636	.631	.468	.430	.055	.661	.670	.597	.567	.072	.710	.732
	EDN [37]	43	.573	.542	.109	.666	.680	.595	.562	.044	.727	.756	.688	.660	.063	.761	.795
	ICON [44]	32	.520	.481	.125	.641	.648	.540	.502	.053	.695	.715	.631	.596	.076	.724	.752
COD	SINetV2 [8]	27	.590	.562	.102	.681	.694	.609	.577	.043	.729	.763	.662	.639	.066	.740	.769
	PFNet [26]	47	.535	.505	.110	.652	.661	.556	.524	.049	.699	.730	.660	.633	.068	.737	.769
	ZoomNet [27]	33	.494	.472	.113	.635	.612	.520	.496	.048	.488	.671	.596	.576	.074	.708	.706
	FEDER [10]	44	.567	.538	.106	.669	.687	.636	.598	.042	.749	.793	.688	.664	.063	.758	.790
	PRNet [13]	13	.648	.607	.096	.716	.766	.709	.672	.059	.772	.820	.650	.603	.038	.756	.815
	ICEG [11]	100	.728	.697	.066	.769	.820	.735	.708	.051	.786	.840	.645	.610	.035	.753	.807
	CamoFormer [40]	71	.645	.618	.078	.732	.750	.729	.707	.054	.789	.822	.668	.639	.035	.770	.811
	PGT [33]	68	.635	.612	.089	.718	.730	.729	.706	.052	.791	.819	.642	.612	.036	.758	.786
	SAM-Adapter [2]	4.11	.661	.638	.080	.744	.753	.688	.667	.037	.788	.808	.727	.710	.051	.794	.809
	SAM2-Adapter [3]	4.36	.717	.692	.074	.779	.807	.724	.694	.044	.809	.847	.735	.694	.045	.819	.845
Unified	VSCode [23]	60	.562	.532	.109	.658	.678	.626	.591	.043	.744	.787	.684	.662	.067	.753	.783
	EVP [20]	4.95	.636	.637	.085	.701	.718	.693	.694	.040	.742	.775	.615	.614	.069	.724	.749
	USCNet(Ours)	**4.04**	**.829**	**.790**	**.049**	**.845**	**.886**	**.794**	**.768**	**.039**	**.839**	**.877**	**.743**	**.700**	**.030**	**.821**	**.869**

numbers of objects in USC12K, as shown in the Table 6.

Detail of annotation process. For Scene A and B, we retained their original annotations, while Scene D did not require additional annotation. Therefore, we focus here on detailing the annotation process for Scene C.

- **Initial Determination of Object Aspects:** We invited 7 observers to perform the initial identification of salient and camouflaged objects in the images. A voting process was used to determine the salient and camouflaged objects in each image, with objects and their aspects receiving more than half of the votes being retained. We then used Photoshop to apply red boxes for salient objects and green boxes for camouflaged objects, which served as the reference for the subsequent mask annotation step.
- **Mask Annotation:** We invited 9 volunteers to perform detailed mask annotation for the dataset using the ISAT interactive annotation tool [14], which supports SAM semi-automatic labeling.

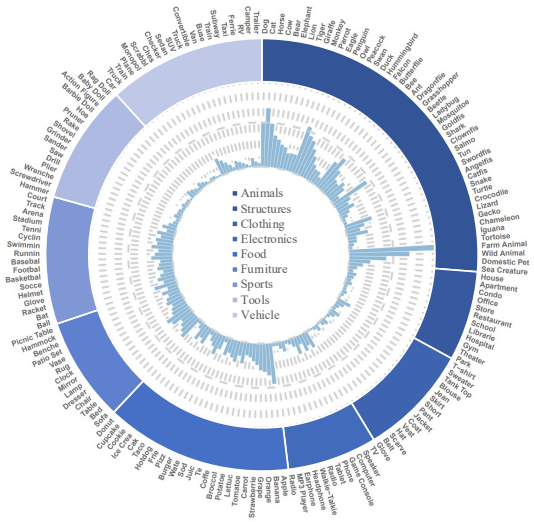


Figure 3. The data source and distribution of different data types.

- **Annotation Quality Control:** After annotation, we invited an additional 3 observers to review and refine the results. Masks with imprecise or incorrect annotations were manually corrected.

More USC12K examples. In Figure 4, we illustrate a selection of images from the USC12K dataset, each featuring both salient and camouflaged objects. The main difference between our USC12K dataset and existing SOD and COD datasets is that it includes a curated subset of 3,000 images, each featuring both salient and camouflaged objects. We invest significant time and effort in finding and annotating these images. Our dataset spans an extensive variety of environments, including, but not limited to, terrestrial, aquatic, alpine, sylvan, and urban ecosystems, and encompasses a broad spectrum of categories, such as lion, flower and various fruit species. This dataset is designed to assist the SOD and COD research communities in advancing the state-of-the-art in discerning more sophisticated saliency and camouflage patterns.

7. Additional Qualitative Results

We present additional predictive results of our USCNet model compared to other COD and SOD models in the USC12K test set. As illustrated in Figure 5, our model outperforms its competitors. Specifically, across four different scenes, our model demonstrates a high degree of consistency with the ground truth, especially in distinguishing between salient and camouflaged objects. Our model is adept at learning distinctive features of saliency and camouflage. For instance, it can accurately identify patterns such as camouflaged humans (refer to the fifth column of Figure 5). Moreover, in scenes devoid of salient or camouflaged objects, our model remains unaffected by complex backgrounds (refer to the sixth column of Figure 5). This further underscores the robustness and accuracy of our USCNet model.

Table 6. Distribution of Images with Different Numbers of Objects in USC12K.

Number of objects	**0**	**1**	**2**	**>2**
Number of images	3000	4197	2335	2468

Table 7. Performance of different base models. *In the original SAM or SAM2, we only fine-tune the mask decoder.

Method	Base	Para.	IoU_S	IoU_C	mIoU	mAcc	CSCS
SAM*	SAM	3.92	51.07	33.00	59.56	68.73	18.66
USCNet	SAM	4.08	73.93	56.50	75.87	83.86	8.24
SAM2*	SAM2	4.22	66.42	44.02	68.78	77.65	11.58
USCNet	SAM2	4.04	**75.57**	**61.34**	**78.03**	**87.92**	**7.49**

8. Additional Ablation Study

Performance of Different Base Models. We conducted ablation experiments to evaluate the performance of different base models, as presented in Table 7. First, as shown in the first two and last two rows of the table, our model demonstrates significant performance improvements on the USC12K benchmark, regardless of whether SAM [15] (default vit-huge version) or SAM2 [30] (default hiera-large version) is used as the base model. For instance, when using SAM as the base model, our method achieves a 16.31% gain in mIoU compared to the original SAM, while utilizing SAM2 results in a 9.25% improvement in mIoU over the original SAM2. Additionally, transitioning from SAM to SAM2 (as shown in rows 2 and 4) results in performance gains across all metrics with fewer fine-tuned parameters.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009. 2

[2] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *ICCVW*, pages 3359–3367, 2023. 4

[3] Tianrun Chen, Ankang Lu, Lanyun Zhu, Chaotao Ding, Chunan Yu, Deyi Ji, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam2-adapter: Evaluating & adapting segment anything 2 in downstream tasks: Camouflage, shadow, medical image segmentation, and more. *arXiv preprint arXiv:2408.04579*, 2024. 2, 3, 4

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

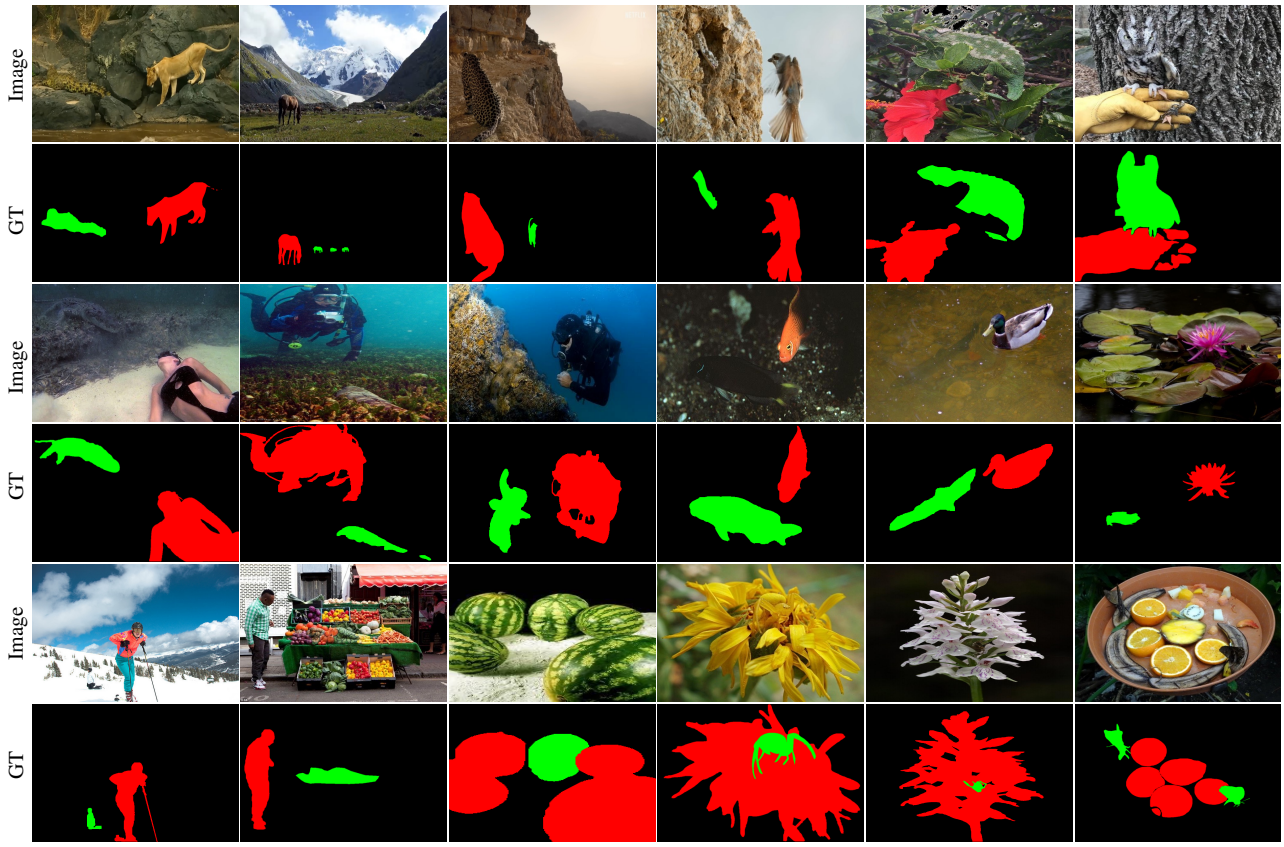


Figure 4. Additional Example images where exist both camouflaged and salient objects from the USC12K dataset. Our collection comprises 3,000 carefully curated and annotated images, encompassing a diverse range of scenes and categories. **Please zoom in for an better view**.

[5] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 2

[6] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *IJCAI*, pages 1–10, 2018. 2

[7] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, pages 2777–2787, 2020. 2, 3

[8] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 44(10): 6024–6042, 2021. 2, 3, 4

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

[10] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Camouflaged object detection with feature decomposition and edge reconstruction. In *CVPR*, pages 22046–22055, 2023. 3, 4

[11] Chunming He, Kai Li, Yachao Zhang, Yulun Zhang, Chenyu You, Zhenhua Guo, Xiu Li, Martin Danelljan, and Fisher Yu. Strategic preys make acute predators: Enhancing camouflaged object detectors by generating camouflaged objects. In *ICLR*, 2024. 3, 4

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[13] Xihang Hu, Xiaoli Zhang, Fasheng Wang, Jing Sun, and Fuming Sun. Efficient camouflaged object detection network based on global localization perception and local guidance refinement. *IEEE TCSVT*, 2024. 3, 4

[14] Shuwei Ji and Hongyuan Zhang. ISAT with Segment Anything: An Interactive Semi-Automatic Annotation Tool, 2023. Updated on 2023-06-03. 4

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 2, 5

[16] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CVIU*, 184:45–56, 2019. 2

[17] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *CVPR*, 2015. 2

[18] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *ICCV*, pages 6015–6026, 2023. 2

[19] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, pages 4722–4732, 2021. 3, 4

[20] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations. In *CVPR*, pages 19434–19445, 2023. 3, 4

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2

[23] Ziyang Luo, Nian Liu, Wangbo Zhao, Xuguang Yang, Dingwen Zhang, Deng-Ping Fan, Fahad Khan, and Junwei Han. Vscode: General visual salient and camouflaged object detection with 2d prompt learning. In *CVPR*, pages 17169–17180, 2024. 3, 4

[24] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, pages 11591–11601, 2021. 2

[25] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, pages 248–255, 2014. 2

[26] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, pages 8772–8781, 2021. 3, 4

[27] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *CVPR*, pages 2160–2170, 2022. 3, 4

[28] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012. 2

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 3

[30] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025. 2, 5

[31] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 2

[32] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 2

[33] Rui Wang, Caijuan Shi, Changyu Duan, Weixiang Gao, Hongli Zhu, Yunchao Wei, and Meiqin Liu. Camouflaged object segmentation with prior via two-stage training. *CVIU*, 246:104061, 2024. 3, 4

[34] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE TPAMI*, 44(6):3239–3259, 2021. 2

[35] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022. 2

[36] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: fusion, feedback and focus for salient object detection. In *AAAI*, pages 12321–12328, 2020. 3, 4

[37] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE TIP*, 31:3125–3136, 2022. 3, 4

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 34:12077–12090, 2021. 2

[39] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 2

[40] Bowen Yin, Xuying Zhang, Deng-Ping Fan, Shaohui Jiao, Ming-Ming Cheng, Luc Van Gool, and Qibin Hou. Camoformer: Masked separable attention for camouflaged object detection. *IEEE TPAMI*, 2024. 3, 4

[41] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, pages 558–567, 2021. 2

[42] Miao Zhang, Tingwei Liu, Yongri Piao, Shunyu Yao, and Huchuan Lu. Auto-msfnet: Search multi-scale fusion network for salient object detection. In *ACM MM*, pages 667–676, 2021. 3, 4

[43] Xiaoqi Zhao, Youwei Pang, Lihe Zhang, Huchuan Lu, and Lei Zhang. Suppress and balance: A simple gated network for salient object detection. In *ECCV*, pages 35–51, 2020. 3, 4

[44] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning. *IEEE TPAMI*, 45(3):3738–3752, 2022. 3, 4

Figure 5. Additional visualizations of the proposed USCNet and other state-of-the-art methods on the USC12K test set. **Zoom-in for better view.**