HDPNet: Hourglass Vision Transformer with Dual-Path Feature Pyramid for Camouflaged Object Detection - Supplementary Materials

Jinpeng He[1], Biyuan Liu[1], Huaixin Chen[1,*]
[1]University of Electronic Science and Technology of China, Chengdu, China
jphe@std.uestc.edu.cn, byliu90@outlook.com, huaixinchen@uestc.edu.cn

Appendices

We will introduce more details that cannot be expanded in the main text.

A. Evaluation Metrics

Following [6, 9], we adopt six well-known evaluation metrics, including S-measure [2] (S_m), weighted F-measure [12] (F_ω), mean F-measure [1] (F_m), mean E-measure [3] (E_m), max E-measure (E_x), and mean absolute error [15] (M). In addition, we plot the precision-recall (PR), Fm-threshold and Em-threshold curves.

S-measure is used to evaluate region-aware (S_r) and object-aware (S_o) structural similarity between predictions and GT and defined as:

$$S = \alpha S_o + (1-\alpha)S_r \tag{1}$$

Where α is set to 0.5.

F-measure is a holistic metric that considers both precision (P) and recall (R), which is defined as:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R} \tag{2}$$

where β is the balance parameter and β^2 is set as 0.3.

Weighted F-measure is derived by combining the weighted precision defined by measure exactness and the weighted recall defined by measure completeness on the basis of the F-measure, calculated as:

$$F_\beta^\omega = \frac{(\beta^2+1)P^\omega R^\omega}{\beta^2 P^\omega + R^\omega} \tag{3}$$

E-measure is used to measure pixel-level matching and image-level statistics,which is defined as:

$$E = \frac{1}{N}\sum_{i=1}^{N} \phi FM(i) \tag{4}$$

where denotes the enhanced-alignment matrix and N is the total pixels of the image.

Mean absolute error. is to calculate the average absolute error of the prediction of camouflaged objects (P) and ground truth (G), which is defined as:

$$M = \frac{1}{N}\sum_{i=1}^{N} |P(i) - G(i)| \tag{5}$$

B. More Comparisons

B.1. Quantitative Comparison

We show more quantitative experimental results on three benchmark COD datasets. The methods used in the experiments for comparison include 7 CNN-based and 7 Transformer-based methods: JCOD [8], BGNet [16], ZoomNet [13], SINetv2 [4], FEDER [5], ZoomNext [14], DGNet [7], UGTR [17], ICON [20], DTINet [11], TPRNet [19], CamoFormer [18], EVP [10] and FSPNet [6].

Quantitative Curves. As illustrated in Fig. 1, for a more intuitive performance evaluation, we plot the precision-recall curves (first row), E-measure curves (second row), and F-measure curves (third row) of our proposed method with 11 top-performing competitors at different thresholds on the three benchmark datasets. All comparisons show that our method (red solid line) significantly outperforms the other methods.

Evaluation for COD10K Super-classes. In addition to the overall quantitative comparison of the COD10K dataset, we also report quantitative results of some representative competitors on each super-class in Tab. 1. We report S-measure (S_m), Weighted F-measurea (F_ω), dapative F-measure (F_m^a), mean F-measure (F_m^m), maximum F-measure (F_m^x), dapative E-measure (E_m^a), mean E-measure (E_m^m), maximum E-measure (E_m^x) and mean absolute error ($\mathcal{M}$) in our experiments. The "adaptive" means that two times the average value of the prediction map pixels is used as the threshold for calculating precision and recall. It can be seen that our model outperforms all competitors across all metrics for each super-class in the COD10K dataset, highlighting its capability to detect the fine structure of camouflaged objects.

Evaluation for COD10K Sub-classes. To investigate the pros and cons of our model, we also provide further quantitative results for selected representative competitors on each sub-class, as shown in Tab. 2. It can be seen that our model outperforms most competitors across S_m metric for each sub-class in the COD10K dataset.

B.2. Qualitative Comparison

Due to the limited space in the manuscript, we add more visual comparisons to this supplementary material for further demonstration of the performance of our model. Figs. 2 to 6 show examples containing small, boundary indistinguishable, occluded, multiple and large camouflaged objects, respectively. As can be seen from these visual comparisons, our method is able to capture the entire object area, recover and preserve edge details, providing more accurate, complete, vivid, and exact high-quality segmentation maps of camouflaged objects with excellent robustness.

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, 2009. 1

[2] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps, 2017. 1

[3] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 698–704. International Joint Conferences on Artificial Intelligence Organization, 7 2018. 1

[4] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6024–6042, Oct. 2022. 1

[5] 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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22046–22055, 2023. 1

[6] Zhou Huang, Hang Dai, Tian-Zhu Xiang, Shuo Wang, Huai-Xin Chen, Jie Qin, and Huan Xiong. Feature shrinkage pyramid for camouflaged object detection with transformers, 2023. 1

[7] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Deep gradient learning for efficient camouflaged object detection. *Machine Intelligence Research*, 20(1):92–108, Jan. 2023. 1

[8] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai. Uncertainty-aware joint salient object and camouflaged object detection, 2021. 1

[9] Yanhua Liang, Guihe Qin, Minghui Sun, Xinchao Wang, Jie Yan, and Zhonghan Zhang. A systematic review of image-level camouflaged object detection with deep learning. *Neurocomputing*, 566:127050, 2024. 1

[10] Weihuang Liu, Xi Shen, Chi-Man Pun, and Xiaodong Cun. Explicit visual prompting for low-level structure segmentations, 2023. 1

[11] Zhengyi Liu, Zhili Zhang, and Wei Wu. Boosting camouflaged object detection with dual-task interactive transformer, 2022. 1

[12] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014. 1

[13] 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, 2022. 1

[14] Youwei Pang, Xiaoqi Zhao, Tian-Zhu Xiang, Lihe Zhang, and Huchuan Lu. Zoomnext: A unified collaborative pyramid network for camouflaged object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[15] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012. 1

[16] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection, 2022. 1

[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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4146–4155, 2021. 1

[18] Bowen Yin, Xuying Zhang, Qibin Hou, Bo-Yuan Sun, Deng-Ping Fan, and Luc Van Gool. Camoformer: Masked separable attention for camouflaged object detection, 2022. 1

[19] Qiao Zhang, Yanliang Ge, Cong Zhang, and Hongbo Bi. Tprnet: camouflaged object detection via transformer-induced progressive refinement network. *The Visual Computer*, 39(10):4593–4607, Oct 2023. 1

[20] Mingchen Zhuge, Deng-Ping Fan, Nian Liu, Dingwen Zhang, Dong Xu, and Ling Shao. Salient object detection via integrity learning, 2022. 1

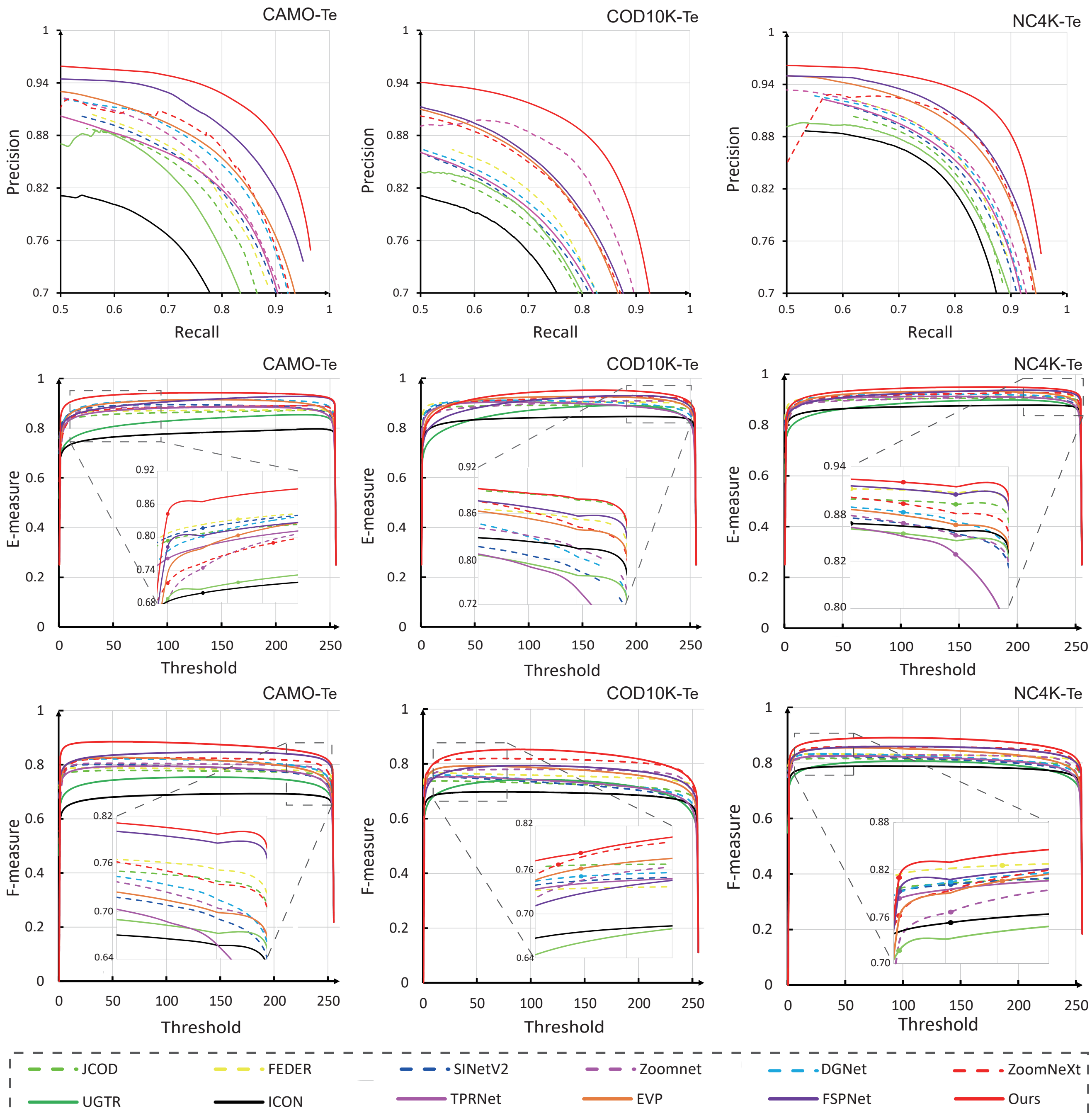

Figure 1. The precision-recall curves (first row), E-measure curves (second row) and F-measure curves (third row) of our method with 11 competitors on three benchmark datasets. The dashed line represents the CNN-based method and the solid line represents the Transformer-based method. The closer the PR curve is to the upper-right corner, the better the performance is. The higher the F-measure/E-measure curve is, the better the model performance.

Table 1. Quantitative comparison of the proposed method with 12 state-of-the-art methods on each super-class of COD10K. The best three results are highlighted in red, green and blue. "↑": the higher the better, "↓": the lower the better.

Super class	Metric	CNN-based Models						Transformer-based Models						
		BGNet	FEDER	SINetV2	ZoomNet	ZoomNext	DGNet	UGTR	ICON	DTINet	TPRNet	EVP	FSPNet	Ours
Aquatic	$S_m \uparrow$	0.816	0.820	0.811	0.830	0.854	0.825	0.812	0.757	0.832	0.811	0.844	0.851	0.885
	$F_\omega \uparrow$	0.719	0.733	0.696	0.735	0.773	0.721	0.689	0.631	0.733	0.697	0.743	0.763	0.808
	$F_m^a \uparrow$	0.741	0.767	0.712	0.753	0.787	0.738	0.708	0.687	0.746	0.716	0.733	0.764	0.793
	$F_m^m \uparrow$	0.758	0.775	0.738	0.775	0.810	0.760	0.737	0.686	0.766	0.742	0.779	0.800	0.835
	$F_m^x \uparrow$	0.776	0.788	0.768	0.791	0.826	0.785	0.764	0.696	0.791	0.771	0.813	0.823	0.863
	$E_m^a \uparrow$	0.883	0.892	0.864	0.881	0.906	0.885	0.860	0.831	0.888	0.864	0.886	0.905	0.917
	$E_m^m \uparrow$	0.887	0.890	0.883	0.880	0.903	0.896	0.857	0.810	0.900	0.877	0.904	0.902	0.924
	$E_m^x \uparrow$	0.895	0.895	0.900	0.897	0.919	0.907	0.887	0.826	0.911	0.893	0.925	0.927	0.944
	$\mathcal{M} \downarrow$	0.049	0.046	0.051	0.044	0.037	0.043	0.050	0.073	0.043	0.052	0.039	0.036	0.027
Terrestrial	$S_m \uparrow$	0.802	0.788	0.787	0.811	0.837	0.796	0.789	0.754	0.794	0.789	0.815	0.824	0.866
	$F_\omega \uparrow$	0.665	0.651	0.623	0.673	0.724	0.641	0.607	0.590	0.636	0.626	0.671	0.679	0.745
	$F_m^a \uparrow$	0.685	0.679	0.622	0.684	0.724	0.645	0.609	0.613	0.642	0.633	0.679	0.688	0.723
	$F_m^m \uparrow$	0.697	0.690	0.662	0.713	0.758	0.676	0.653	0.630	0.668	0.668	0.709	0.716	0.774
	$F_m^x \uparrow$	0.716	0.705	0.697	0.730	0.777	0.706	0.686	0.645	0.696	0.697	0.736	0.743	0.812
	$E_m^a \uparrow$	0.881	0.880	0.835	0.863	0.887	0.855	0.820	0.827	0.855	0.842	0.886	0.880	0.892
	$E_m^m \uparrow$	0.876	0.878	0.866	0.863	0.886	0.875	0.822	0.816	0.873	0.864	0.883	0.871	0.901
	$E_m^x \uparrow$	0.890	0.884	0.888	0.892	0.911	0.893	0.871	0.827	0.891	0.887	0.907	0.919	0.942
	$\mathcal{M} \downarrow$	0.035	0.033	0.039	0.030	0.028	0.035	0.036	0.052	0.037	0.039	0.031	0.027	0.022
Flying	$S_m \uparrow$	0.862	0.852	0.839	0.865	0.885	0.840	0.843	0.806	0.844	0.844	0.866	0.871	0.906
	$F_\omega \uparrow$	0.769	0.759	0.713	0.766	0.801	0.716	0.700	0.675	0.721	0.718	0.764	0.761	0.821
	$F_m^a \uparrow$	0.779	0.775	0.710	0.771	0.797	0.714	0.697	0.702	0.723	0.721	0.766	0.758	0.797
	$F_m^m \uparrow$	0.796	0.790	0.749	0.800	0.832	0.749	0.744	0.719	0.752	0.756	0.797	0.793	0.845
	$F_m^x \uparrow$	0.820	0.810	0.786	0.817	0.851	0.785	0.776	0.732	0.783	0.788	0.825	0.818	0.879
	$E_m^a \uparrow$	0.932	0.925	0.888	0.908	0.925	0.895	0.870	0.869	0.900	0.886	0.924	0.913	0.934
	$E_m^m \uparrow$	0.931	0.924	0.908	0.909	0.925	0.914	0.873	0.855	0.916	0.906	0.927	0.910	0.938
	$E_m^x \uparrow$	0.942	0.930	0.924	0.934	0.942	0.931	0.912	0.869	0.931	0.925	0.946	0.941	0.965
	$\mathcal{M} \downarrow$	0.021	0.022	0.027	0.020	0.018	0.026	0.026	0.035	0.026	0.026	0.022	0.020	0.015
Amphibian	$S_m \uparrow$	0.873	0.857	0.858	0.874	0.878	0.854	0.857	0.845	0.851	0.858	0.877	0.878	0.911
	$F_\omega \uparrow$	0.790	0.774	0.756	0.784	0.797	0.751	0.738	0.752	0.741	0.754	0.790	0.782	0.841
	$F_m^a \uparrow$	0.810	0.794	0.761	0.798	0.804	0.764	0.749	0.782	0.757	0.768	0.808	0.786	0.837
	$F_m^m \uparrow$	0.815	0.803	0.788	0.811	0.825	0.780	0.774	0.788	0.768	0.788	0.820	0.809	0.864
	$F_m^x \uparrow$	0.836	0.814	0.819	0.829	0.839	0.807	0.794	0.798	0.790	0.816	0.846	0.830	0.889
	$E_m^a \uparrow$	0.933	0.931	0.893	0.913	0.922	0.909	0.882	0.909	0.900	0.899	0.934	0.927	0.951
	$E_m^m \uparrow$	0.930	0.930	0.916	0.913	0.915	0.912	0.895	0.897	0.905	0.918	0.923	0.921	0.943
	$E_m^x \uparrow$	0.939	0.936	0.937	0.929	0.933	0.925	0.911	0.909	0.919	0.939	0.937	0.948	0.961
	$\mathcal{M} \downarrow$	0.026	0.027	0.030	0.026	0.024	0.028	0.030	0.037	0.034	0.031	0.025	0.023	0.018
Other	$S_m \uparrow$	0.807	0.805	0.779	0.847	0.914	0.831	0.810	0.877	0.783	0.799	0.889	0.899	0.933
	$F_\omega \uparrow$	0.659	0.599	0.589	0.690	0.831	0.672	0.635	0.770	0.599	0.639	0.776	0.789	0.854
	$F_m^a \uparrow$	0.663	0.632	0.588	0.599	0.688	0.622	0.584	0.735	0.589	0.614	0.688	0.672	0.729
	$F_m^m \uparrow$	0.675	0.627	0.615	0.706	0.847	0.706	0.668	0.784	0.613	0.666	0.789	0.810	0.869
	$F_m^x \uparrow$	0.694	0.670	0.644	0.744	0.886	0.773	0.708	0.816	0.633	0.691	0.821	0.845	0.901
	$E_m^a \uparrow$	0.793	0.779	0.738	0.749	0.834	0.770	0.732	0.862	0.763	0.749	0.829	0.827	0.866
	$E_m^m \uparrow$	0.800	0.774	0.775	0.895	0.937	0.848	0.814	0.889	0.790	0.795	0.913	0.917	0.947
	$E_m^x \uparrow$	0.808	0.841	0.821	0.954	0.974	0.897	0.856	0.931	0.841	0.806	0.966	0.977	0.992
	$\mathcal{M} \downarrow$	0.036	0.019	0.031	0.011	0.008	0.023	0.032	0.026	0.045	0.034	0.012	0.011	0.008

Table 2. Results of S_m for each sub-class in COD10K. The best and second performing methods of each category are highlighted in **bold** and underlined, respectively.

Superclass	BGNet	FEDER	SINetV2	ZoomNet	ZoomNext	DGNet	UGTR	ICON	DTINet	TPRNet	EVP	FSPNet	Ours
Amphibian-Frog	0.861	0.850	0.857	0.876	0.867	0.843	0.840	0.808	0.848	0.847	0.874	0.873	**0.912**
Amphibian-Toad	0.887	0.874	0.870	0.885	0.895	0.873	0.875	0.868	0.867	0.872	0.887	0.893	**0.914**
Aquatic-BatFish	0.893	0.829	0.873	0.890	**0.925**	0.859	0.860	0.746	0.902	0.863	0.897	0.907	0.919
Aquatic-ClownFish	0.757	0.846	0.787	0.813	0.860	0.844	0.631	0.736	0.819	0.819	0.849	0.851	**0.899**
Aquatic-Crab	0.830	0.835	0.815	0.835	0.867	0.827	0.838	0.792	0.832	0.810	0.852	0.864	**0.886**
Aquatic-Crocodile	0.786	0.842	0.802	0.822	0.876	0.830	0.810	0.692	0.829	0.804	0.874	0.854	**0.877**
Aquatic-CrocodileFish	0.731	0.797	0.746	0.805	0.809	0.794	0.837	0.455	0.822	0.784	**0.907**	0.846	0.901
Aquatic-Fish	0.819	0.828	0.829	0.847	0.874	0.851	0.801	0.754	0.857	0.830	0.872	0.861	**0.898**
Aquatic-Flounder	0.881	0.877	0.889	0.880	0.913	0.887	0.870	0.757	0.879	0.892	0.907	0.922	**0.934**
Aquatic-FrogFish	0.856	0.891	0.894	0.925	**0.929**	0.888	0.845	0.776	0.914	0.847	0.887	0.925	0.911
Aquatic-GhostPipefish	0.823	0.832	0.817	0.849	0.854	0.831	0.832	0.752	0.861	0.837	0.861	0.872	**0.898**
Aquatic-LeafySeaDragon	0.671	0.628	0.670	0.691	0.737	0.606	0.720	0.637	0.711	0.710	0.677	0.782	**0.785**
Aquatic-Octopus	0.901	0.873	0.887	0.889	0.896	0.885	0.895	0.888	0.873	0.907	**0.908**	0.885	0.907
Aquatic-Pagurian	0.691	0.709	0.698	0.724	0.714	0.686	0.740	0.669	0.692	0.660	0.690	0.710	**0.735**
Aquatic-Pipefish	0.811	0.810	0.789	0.816	0.843	0.799	0.807	0.745	0.815	0.794	0.817	0.838	**0.877**
Aquatic-ScorpionFish	0.808	0.807	0.808	0.834	0.869	0.848	0.776	0.703	0.849	0.820	0.862	0.851	**0.884**
Aquatic-SeaHorse	0.809	0.827	0.823	0.825	0.855	0.834	0.826	0.781	0.834	0.815	0.844	0.851	**0.892**
Aquatic-Shrimp	0.762	0.724	0.735	0.786	0.808	0.760	0.741	0.662	0.740	0.731	0.794	0.819	**0.832**
Aquatic-Slug	0.772	**0.861**	0.729	0.776	0.785	0.845	0.752	0.788	0.715	0.750	0.835	0.696	0.785
Aquatic-StarFish	0.869	0.913	0.890	0.892	0.924	0.891	0.850	0.862	0.890	0.863	0.916	0.889	**0.947**
Aquatic-Stingaree	0.757	0.824	0.815	0.817	0.739	**0.891**	0.741	0.720	**0.891**	0.738	0.884	0.881	0.887
Aquatic-Turtle	0.809	0.801	0.760	0.898	0.785	0.877	0.813	0.727	0.897	0.792	0.891	0.883	**0.910**
Flying-Bat	0.857	0.841	0.853	0.838	0.899	0.834	0.873	0.805	0.854	0.847	0.876	0.883	**0.933**
Flying-Bee	0.824	0.780	0.804	0.786	0.827	0.794	0.823	0.787	0.779	0.829	0.818	0.734	**0.891**
Flying-Beetle	0.941	0.847	0.903	0.937	0.941	0.861	0.930	0.920	0.900	0.934	0.902	0.932	**0.947**
Flying-Bird	0.863	0.852	0.835	0.867	0.892	0.841	0.842	0.854	0.849	0.831	0.874	0.873	**0.908**
Flying-Bittern	0.853	0.877	0.849	0.895	0.889	0.869	0.869	0.872	0.873	0.853	0.877	0.865	**0.899**
Flying-Butterfly	0.901	0.899	0.883	0.898	0.904	0.874	0.864	0.847	0.866	0.888	0.892	0.885	**0.924**
Flying-Cicada	0.895	0.891	0.883	0.916	0.924	0.898	0.888	0.874	0.876	0.884	0.906	0.909	**0.941**
Flying-Dragonfly	0.853	0.812	0.837	0.843	0.897	0.821	0.858	0.833	0.846	0.833	0.871	0.886	**0.909**
Flying-Frogmouth	0.948	0.951	0.941	**0.961**	0.938	0.942	0.936	0.866	0.939	0.938	0.954	0.947	0.949
Flying-Grasshopper	0.856	0.847	0.833	0.853	0.878	0.831	0.837	0.796	0.851	0.841	0.859	0.874	**0.901**
Flying-Heron	0.868	0.861	0.823	0.890	**0.901**	0.827	0.826	0.852	0.847	0.859	0.867	0.866	0.899
Flying-Katydid	0.842	0.829	0.809	0.846	0.869	0.811	0.834	0.724	0.814	0.822	0.842	0.853	**0.890**
Flying-Mantis	0.790	0.794	0.775	0.804	0.824	0.783	0.784	0.731	0.779	0.773	0.799	0.835	**0.866**
Flying-Mockingbird	0.895	0.872	0.838	0.863	0.897	0.831	0.777	0.856	0.866	0.841	0.886	0.875	**0.926**
Flying-Moth	0.932	0.921	0.917	0.916	0.947	0.917	0.883	0.828	0.925	0.914	0.925	0.941	**0.959**
Flying-Owl	0.883	0.869	0.868	0.895	0.898	0.860	0.860	0.842	0.856	0.873	0.892	0.875	**0.912**
Flying-Owlfly	0.836	0.857	0.863	0.879	0.885	0.856	0.847	0.693	0.884	0.862	0.879	0.872	**0.902**
Other-Other	0.807	0.805	0.779	0.847	0.914	0.831	0.810	0.877	0.783	0.799	0.889	0.899	**0.933**
Terrestrial-Ant	0.711	0.670	0.669	0.744	**0.768**	0.661	0.715	0.674	0.664	0.721	0.689	0.743	0.749
Terrestrial-Bug	0.870	0.882	0.856	0.900	**0.912**	0.852	0.862	0.806	0.856	0.865	0.875	0.874	0.908
Terrestrial-Cat	0.778	0.759	0.773	0.787	0.819	0.784	0.769	0.711	0.789	0.765	0.811	0.823	**0.866**
Terrestrial-Caterpillar	0.767	0.714	0.776	0.795	0.836	0.777	0.785	0.630	0.805	0.749	0.787	0.813	**0.864**
Terrestrial-Centipede	0.704	0.726	0.762	0.733	0.809	0.795	0.749	0.738	0.718	0.776	0.761	0.791	**0.819**
Terrestrial-Chameleon	0.824	0.825	0.804	0.833	0.850	0.824	0.833	0.801	0.842	0.812	0.839	0.845	**0.865**
Terrestrial-Cheetah	0.843	0.827	0.826	0.821	0.842	0.808	0.826	0.828	0.825	0.830	0.845	0.851	**0.871**
Terrestrial-Deer	0.777	0.740	0.757	0.787	0.790	0.773	0.748	0.789	0.766	0.762	0.801	0.798	**0.842**
Terrestrial-Dog	0.692	0.712	0.707	0.730	0.735	0.719	0.707	0.688	0.774	0.713	0.761	0.786	**0.826**
Terrestrial-Duck	0.751	0.770	0.746	0.740	0.746	0.716	0.724	0.720	0.718	0.713	0.787	0.784	**0.824**
Terrestrial-Gecko	0.857	0.822	0.848	0.856	0.862	0.880	0.840	0.725	0.887	0.866	0.894	0.908	**0.938**
Terrestrial-Giraffe	0.837	0.794	0.784	0.826	0.855	0.826	0.796	0.773	0.847	0.764	0.809	0.846	**0.891**
Terrestrial-Grouse	0.937	0.925	0.921	0.941	**0.959**	0.943	0.947	0.933	0.942	0.931	0.940	0.942	**0.959**
Terrestrial-Human	0.779	0.768	0.817	0.781	0.821	0.782	0.782	0.787	0.806	0.801	0.786	0.797	**0.842**
Terrestrial-Kangaroo	0.806	0.776	0.816	0.800	**0.896**	0.798	0.760	0.849	0.796	0.820	0.794	0.802	0.873
Terrestrial-Leopard	0.836	0.828	0.823	0.848	0.870	0.835	0.858	0.844	0.816	0.842	0.844	0.851	**0.879**
Terrestrial-Lion	0.816	0.826	0.813	0.814	0.857	0.805	0.837	0.805	0.787	0.822	0.848	**0.859**	0.839
Terrestrial-Lizard	0.851	0.833	0.830	0.852	0.876	0.844	0.834	0.835	0.828	0.835	0.852	0.853	**0.895**
Terrestrial-Monkey	0.897	0.874	0.888	0.898	**0.936**	0.768	0.885	0.862	0.811	0.825	0.883	0.913	0.930
Terrestrial-Rabbit	0.860	0.845	0.843	0.854	0.875	0.848	0.838	0.871	0.858	0.842	0.884	0.887	**0.915**
Terrestrial-Reccoon	0.845	0.739	0.766	0.837	**0.878**	0.776	0.726	0.606	0.589	0.784	0.833	0.791	0.839
Terrestrial-Sciuridae	0.883	0.856	0.842	0.897	**0.916**	0.880	0.848	0.864	0.831	0.867	0.897	0.856	0.902
Terrestrial-Sheep	0.481	0.489	0.500	0.504	0.526	0.493	0.490	0.438	0.492	0.676	**0.687**	0.493	0.492
Terrestrial-Snake	0.845	0.845	0.831	0.862	**0.884**	0.834	0.843	0.828	0.832	0.829	0.846	0.854	**0.884**
Terrestrial-Spider	0.794	0.788	0.771	0.812	0.850	0.785	0.779	0.706	0.772	0.775	0.799	0.808	**0.866**
Terrestrial-StickInsect	0.754	0.725	0.696	0.753	0.800	0.727	0.683	0.673	0.703	0.727	0.733	0.762	**0.825**
Terrestrial-Tiger	0.728	0.700	0.703	0.700	0.709	0.710	0.712	0.694	0.679	0.678	0.733	0.734	**0.758**
Terrestrial-Wolf	0.755	0.714	0.749	0.794	0.759	0.717	0.760	0.738	0.741	0.748	0.737	0.749	**0.795**
Terrestrial-Worm	0.800	0.776	0.806	0.807	0.812	0.794	0.837	0.657	0.828	0.774	0.779	0.812	**0.903**

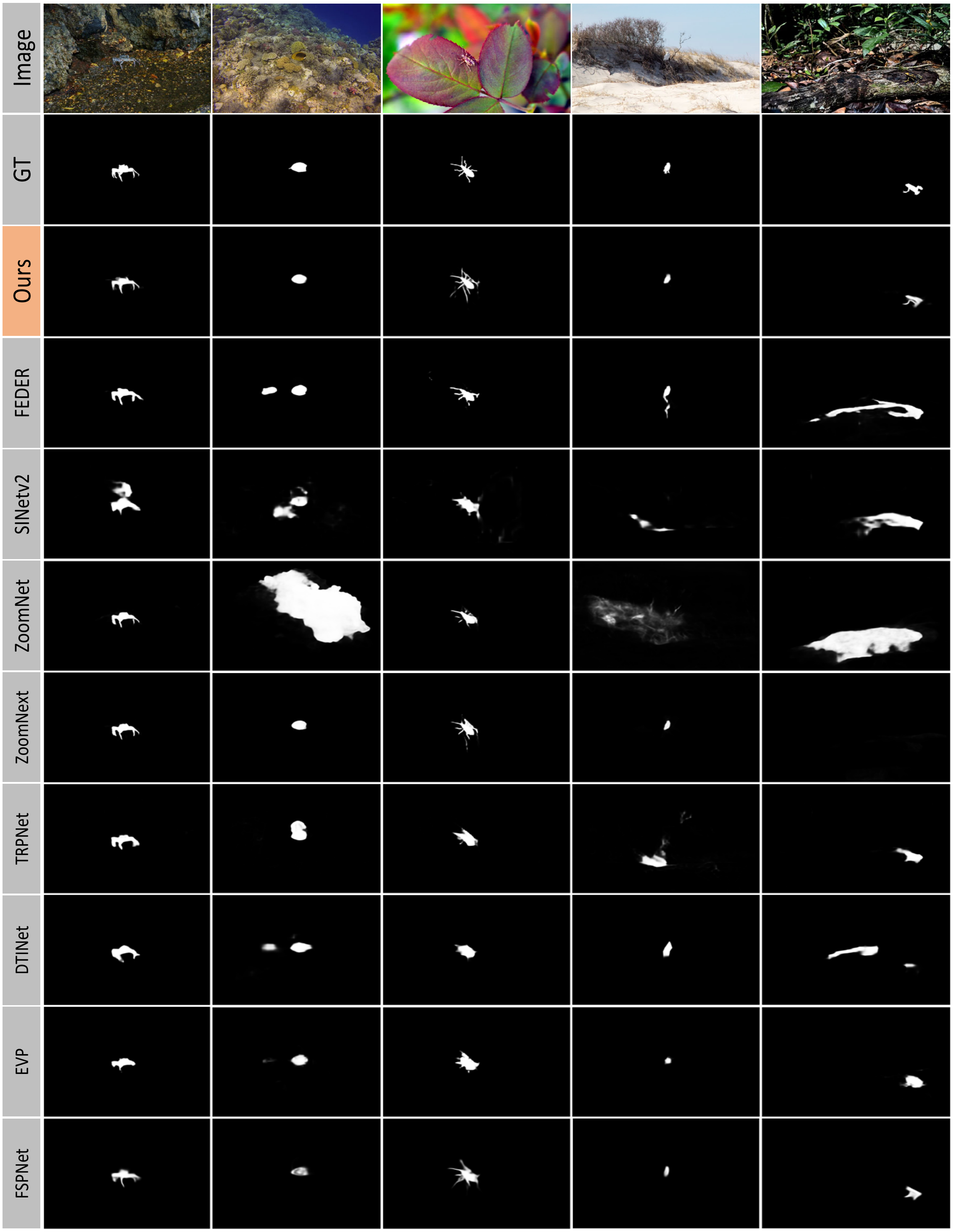

Figure 2. Visual comparison with other competitors in detecting **small** camouflaged objects. Please zoom in for details.

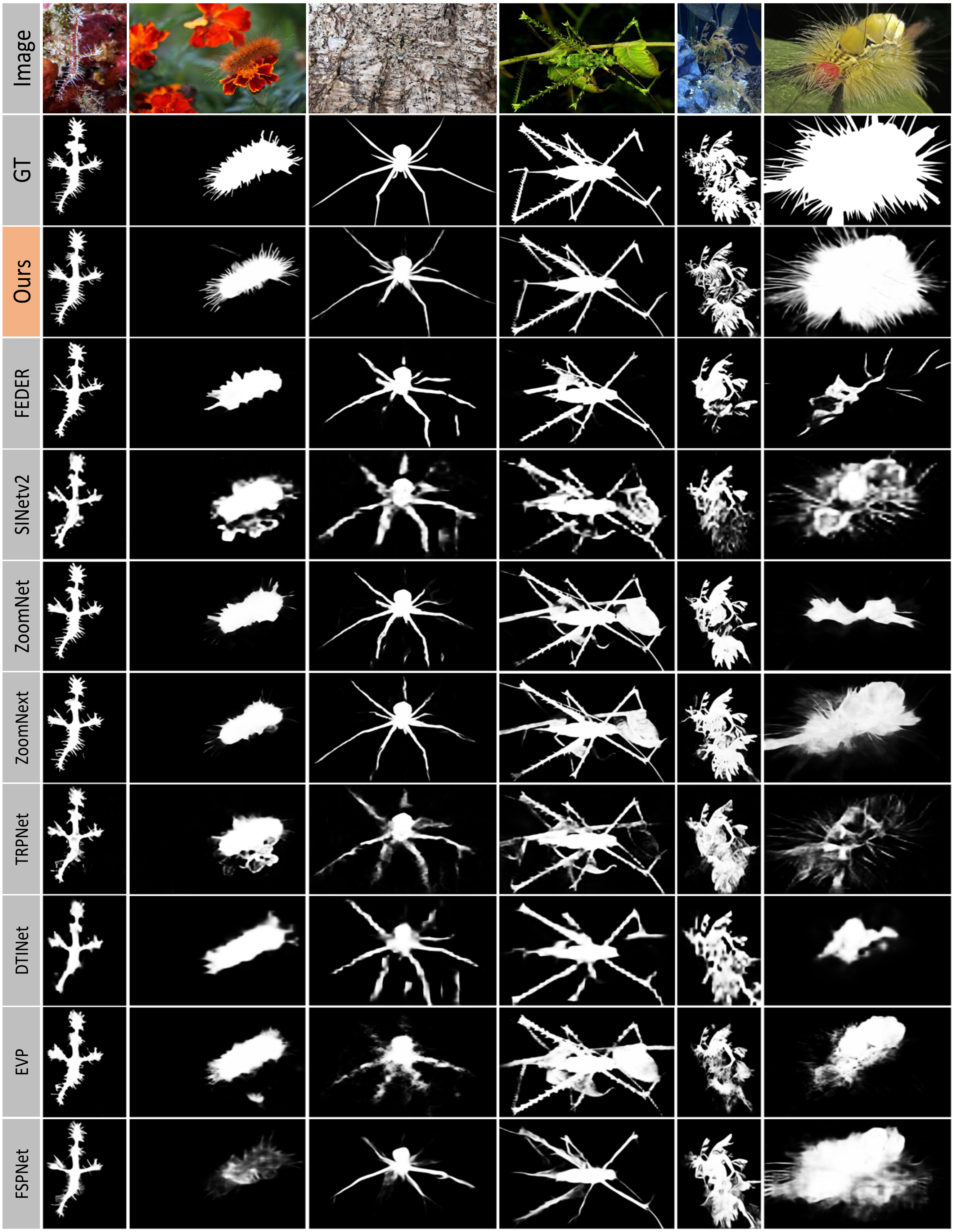

Figure 3. Visual comparison with other competitors in detecting the **detailed boundary** of camouflaged objects. Please zoom in for details.

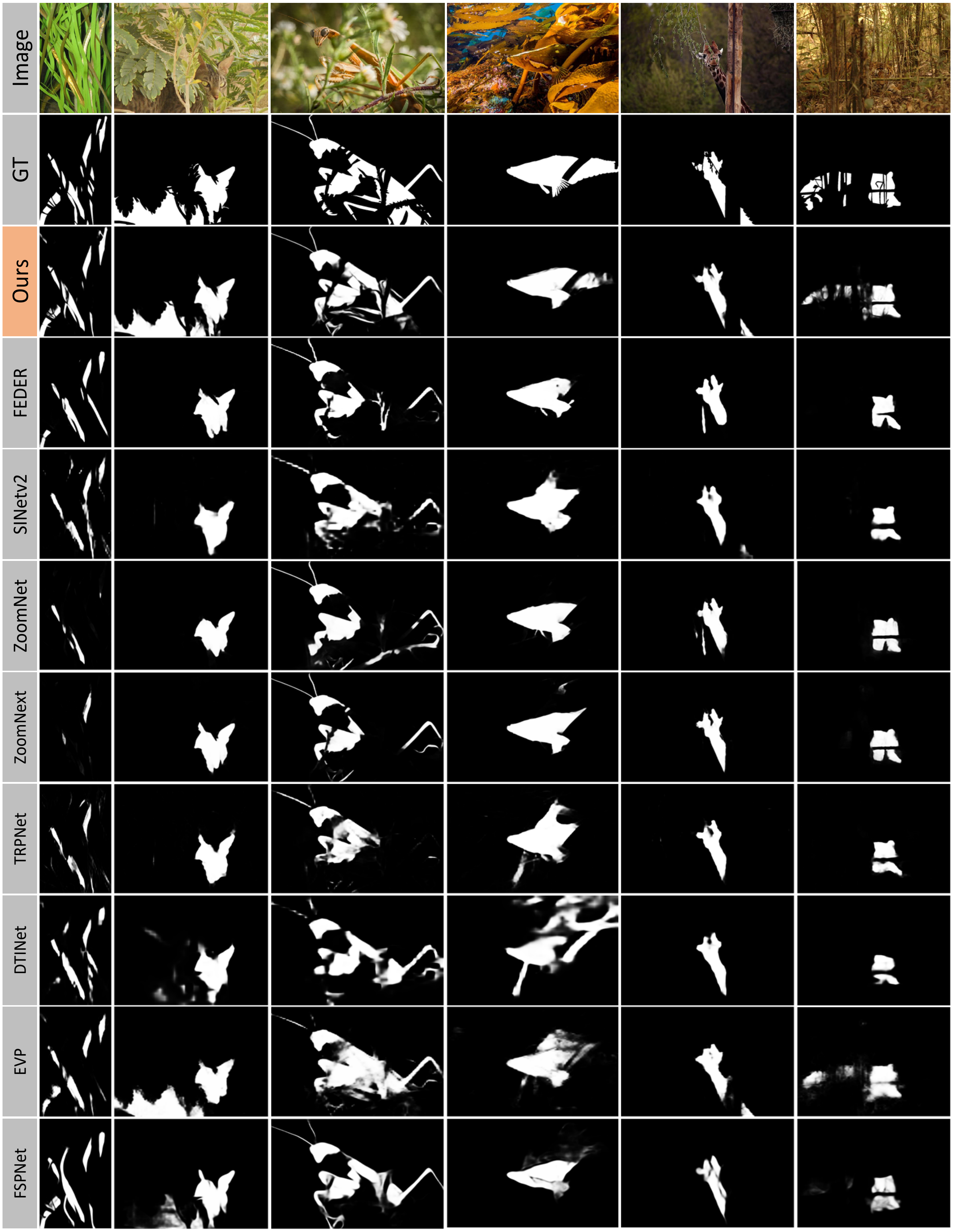

Figure 4. Visual comparison with other competitors in detecting **occluded** camouflaged objects. Please zoom in for details.

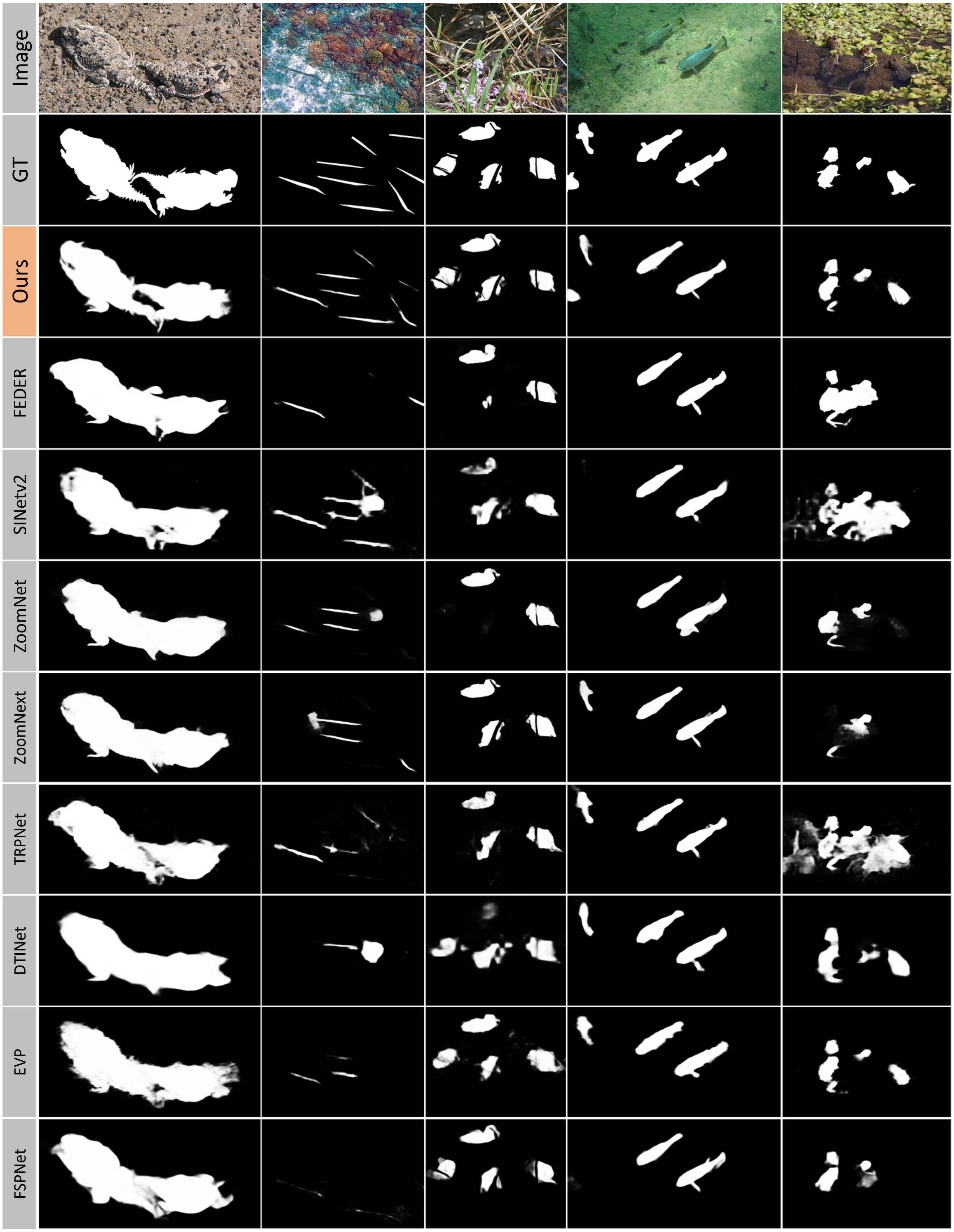

Figure 5. Visual comparison with other competitors in detecting **multiple** camouflaged objects. Please zoom in for details.

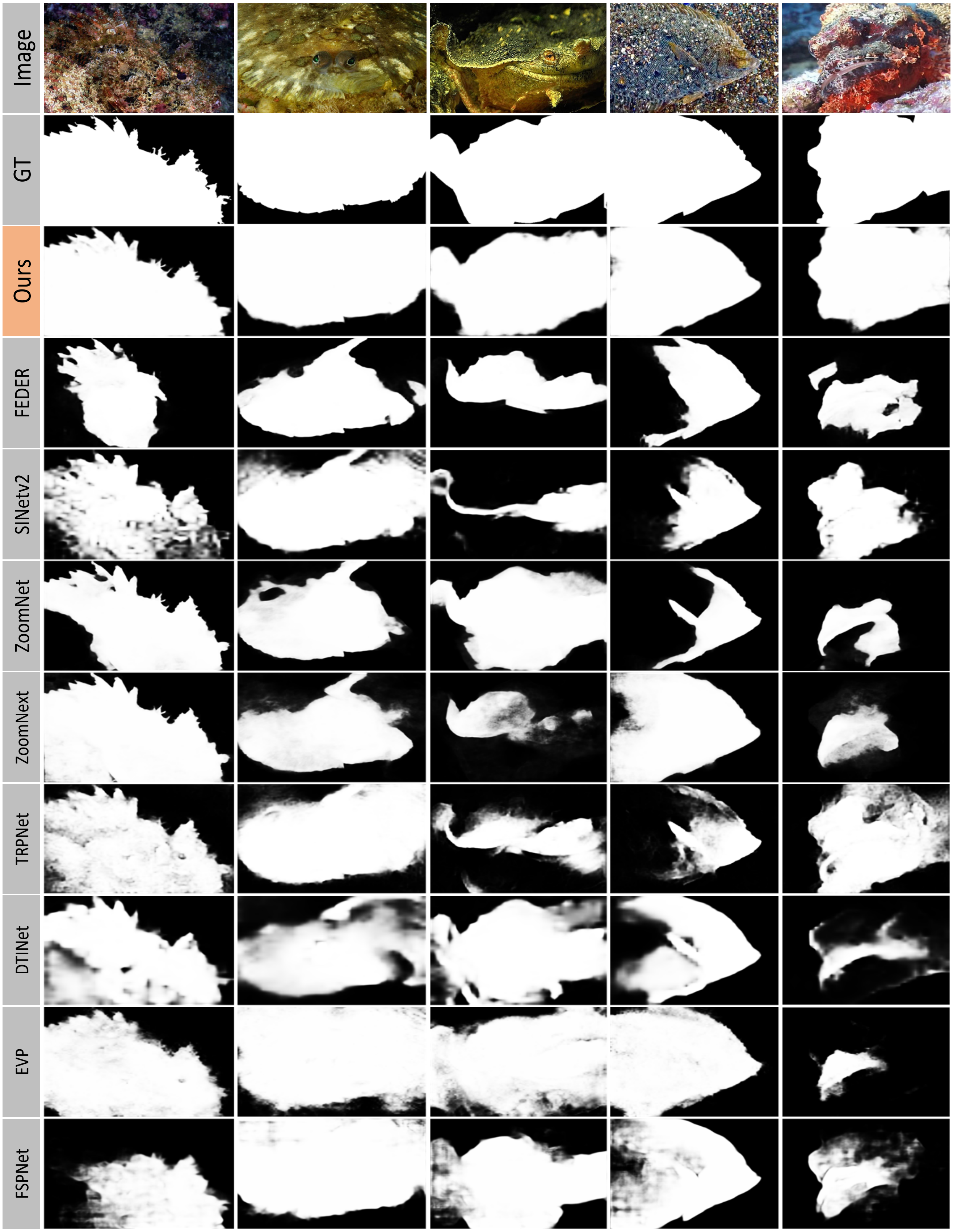

Figure 6. Visual comparison with other competitors in detecting **big** camouflaged objects. Please zoom in for details.