JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Camouflaged Instance Segmentation via Explicit De-camouflaging

Naisong Luo[1*], Yuwen Pan[1*], Rui Sun[1], Tianzhu Zhang[1,2,3], Zhiwei Xiong[1,2], Feng Wu[1,2]

[1]University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3]Deep Space Exploration Lab

Paper Tag： THU-AM-137

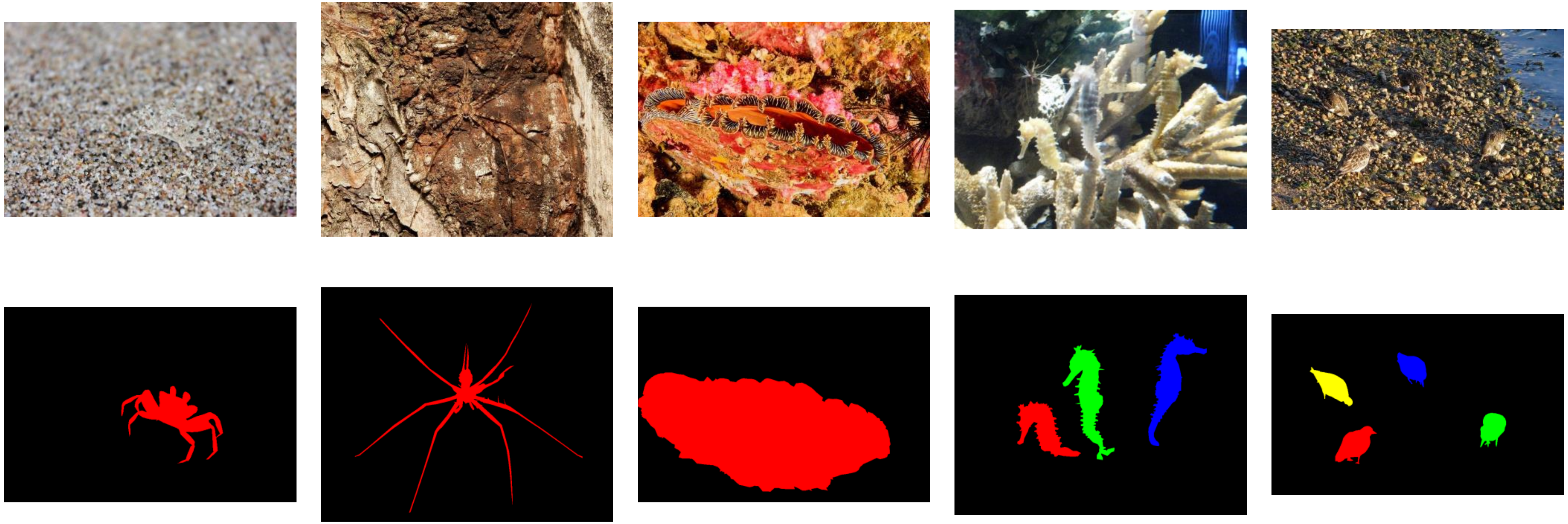
Camouflaged Instance Segmentation via Explicit De-camouflaging

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Camouflaged Instance Segmentation (CIS)

Camouflaged Images

Camouflaged Instances



Core of CIS: De-camouflage

General instance segmentation methods work poorly on this task.

Recent CIS approaches are generally based on traditional instance segmentation models.

→ Because the previous methods are easily disturbed by similar background.

→ The core of CIS is de-camouflaging, *i.e.*, eliminating camouflage characteristics of the target object.

Intuitively, humans first repeatedly **discriminate the real target characteristics** from the camouflage characteristics at the pixel level, and then aggregate the pixel information to **discern the whole target instance** from the background.

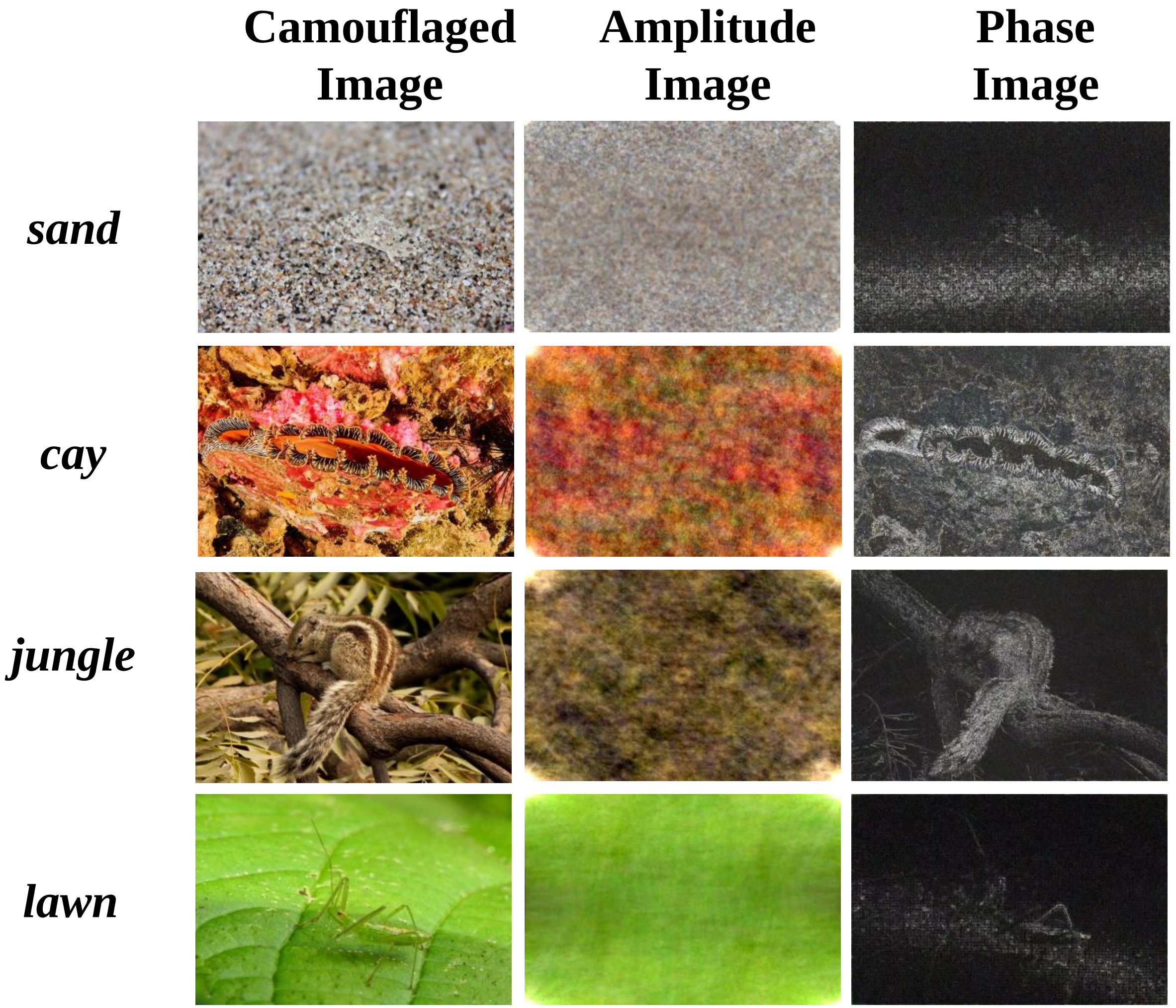
→ Explore decamouflaging strategy from the **pixel level** to the **instance level** in a progressive manner.

DSEL
DEEP SPACE EXPLORATION LAB
深空探测实验室

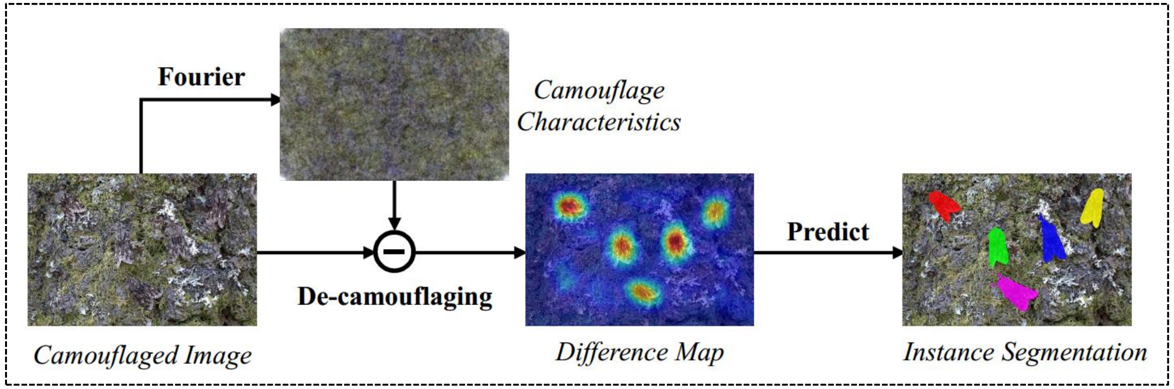
Camouflaged Instance Segmentation via Explicit De-camouflaging

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

Pixel-level De-camouflage



- The Fourier spectrum amplitude contains low-level statistics (e.g., color and texture of the environment) that accords with the camouflage characteristics.
- Although phase images contain semantic information, they also have abundant pixel-level noise (in the background area), which is not conducive to de-camouflaging.

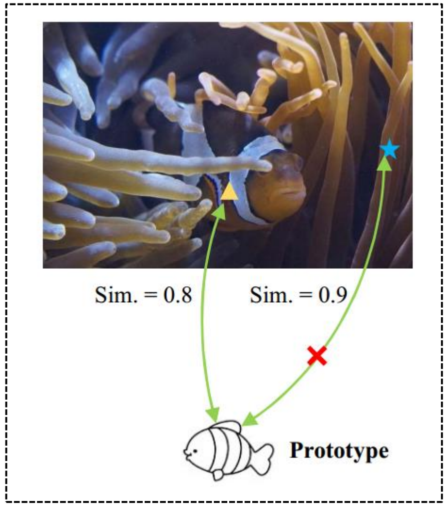


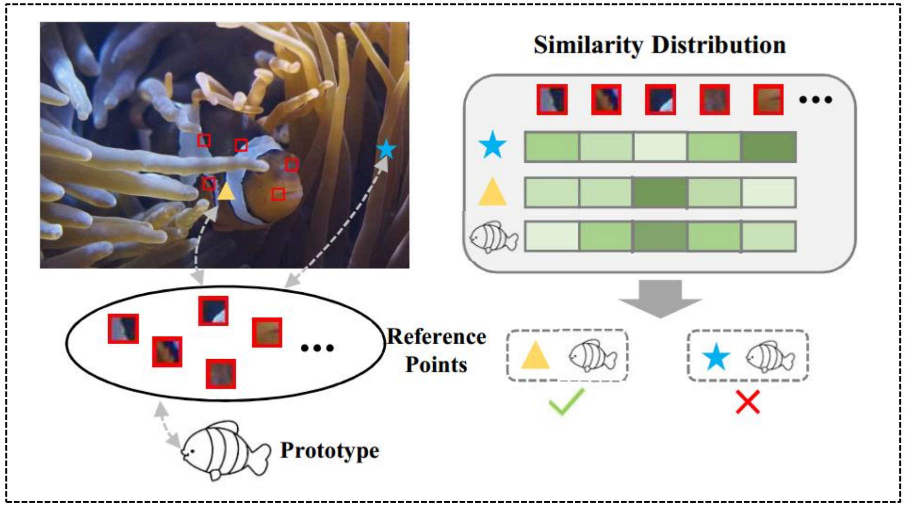
Camouflaged Instance Segmentation via Explicit De-camouflaging

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Instance-level De-camouflage

We introduce a set of instance prototypes to capture each camouflaged instance through long-range context-aware interactions to achieve final instance segmentation.

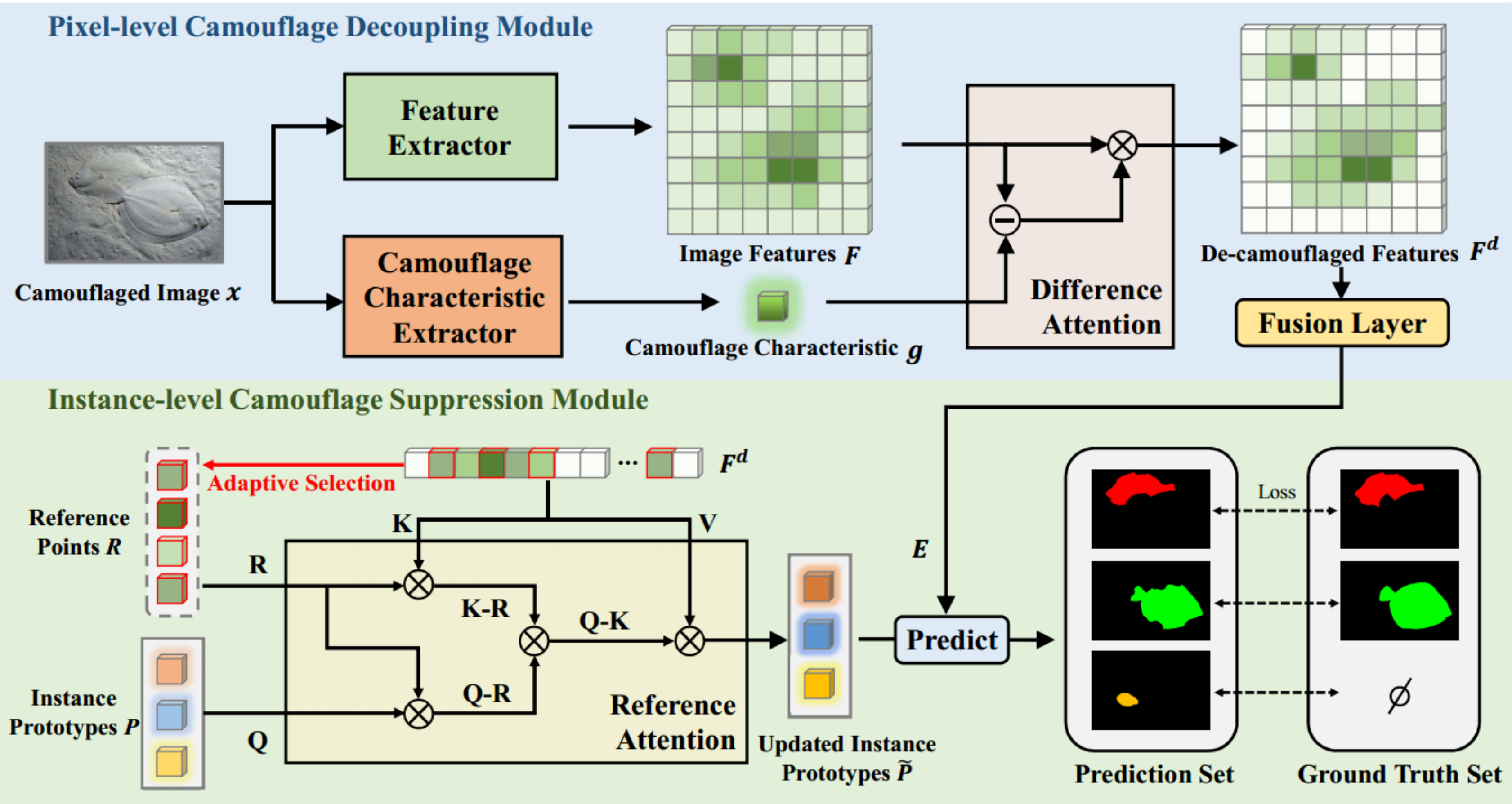




- Prototypes will frequently absorb deceptive background information that has high similarity with the objects during the interaction, thus failing to discover desired targets accurately.
- We select de-camouflaged pixels with high contribution to prototypes as reference points. Highly similar pixels and prototypes must have consistent similarity distributions on the reference points.

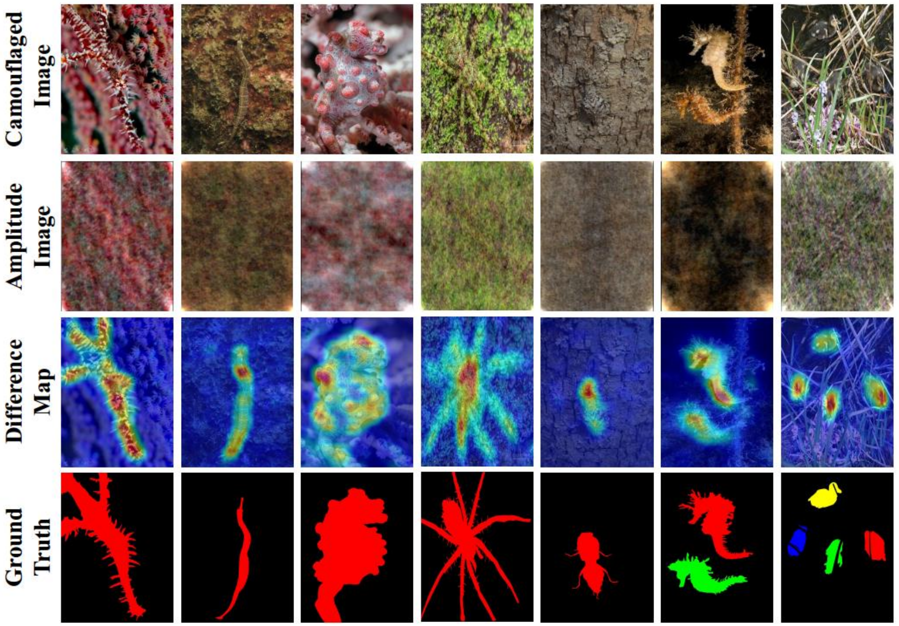
Overall Framework

De-camouflage Network (DCNet)



Camouflaged Instance Segmentation via Explicit De-camouflaging

Pixel-level Camouflage Decoupling Moudule (PCD)



$$\mathcal{F}(x)_{u,v} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j} e^{-J2\pi\left(\frac{i}{H}u + \frac{j}{W}v\right)}$$

$$\mathcal{A}(x)_{u,v} = \left[R^2(x)_{u,v} + I^2(x)_{u,v}\right]^{1/2}$$

$$\mathcal{P}(x)_{u,v} = \arctan\left[\frac{I(x)_{u,v}}{R(x)_{u,v}}\right]$$

Fourier transform

$$\tilde{x} = \mathcal{F}^{-1}\left[\mathcal{A}(x)_{u,v} e^{-Jb}\right]$$

amplitude image

$$\boldsymbol{g} = \text{AvgPooling}\left(\text{CNN}(\tilde{x})\right)$$

global camouflage characteristics

$$\mathbf{D}_{i,j} = \sum_{k=1}^{C} (\hat{\mathbf{F}}_{i,j,k} - \hat{\boldsymbol{g}}_k)^2$$

difference map

Instance-level Camouflage Suppression Moudule (ICS)

$$\boldsymbol{q}_i = \boldsymbol{p}_i \mathbf{W}^q, \boldsymbol{k}_j = \boldsymbol{f}_j \mathbf{W}^k, \boldsymbol{v}_j = \boldsymbol{f}_j \mathbf{W}^v$$

$$s_{i,j} = \frac{d\left(\boldsymbol{q}_i, \boldsymbol{k}_j\right)}{\sqrt{L}}$$ original prototype-pixel similarity

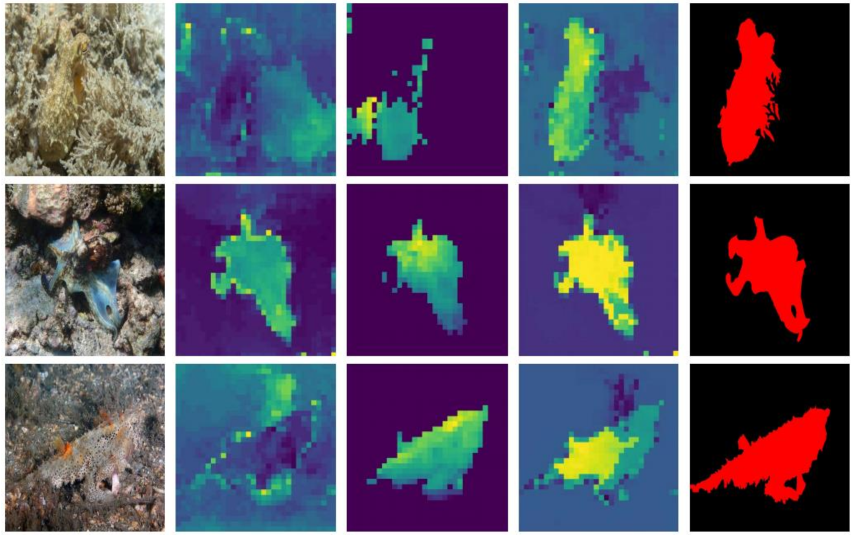
$$u_j = \sum_{i=1}^{N} s_{i,j}, j \in 1, 2, ..., hw$$ contribution to select references

$$s_i^q = \phi_1(\boldsymbol{q}_i \mathbf{R}^\top), s_j^k = \phi_2(\boldsymbol{k}_j \mathbf{R}^\top)$$ prototype-reference and pixel-reference similarity

$$s_{i,j}^{qk} = d(\boldsymbol{q}_i, \boldsymbol{k}_j; \mathbf{R}) = s_i^q (s_j^k)^\top$$ updated prototype-pixel similarity

$$\tilde{\boldsymbol{p}}_i = \sum_{j=1}^{hw} a_{i,j} \boldsymbol{v}_j, a_{i,j} = \frac{\exp(s_{i,j}^{qk})}{\sum_{j=1}^{hw} \exp(s_{i,j}^{qk})}$$ instance prototypes

Reference attention Map



(a) Camouflaged Image (b) Cross-Attention (c) Masked Attention (d) Reference Attention (e) Ground Truth

Camouflaged Instance Segmentation via Explicit De-camouflaging

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Experimental Results

➢ Performance comparison on the COD10K and NC4K dataset

Methods	COD10K-Test			NC4K-Test			Params(M)	GFLOPs
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}		
Mask R-CNN [14]	25.0	55.5	20.4	27.7	58.6	22.7	43.9	186.3
MS R-CNN [17]	30.1	57.2	28.7	31.0	58.7	29.4	60.0	198.5
Cascade R-CNN [2]	25.3	56.1	21.3	29.5	60.8	24.8	71.7	334.1
HTC [4]	28.1	56.3	25.1	29.8	59.0	26.6	76.9	331.7
BlendMask [3]	28.2	56.4	25.2	27.7	56.7	24.2	35.8	233.8
Mask Transfiner [20]	28.7	56.3	26.4	29.4	56.7	27.2	44.3	185.1
YOLACT [1]	24.3	53.3	19.7	32.1	65.3	27.9	-	-
CondInst [38]	30.6	63.6	26.1	33.4	67.4	29.4	34.1	200.1
QueryInst [12]	28.5	60.1	23.1	33.0	66.7	29.4	-	-
SOTR [13]	27.9	58.7	24.1	29.3	61.0	25.6	63.1	476.7
SOLOv2 [41]	32.5	63.2	29.9	34.4	65.9	31.9	46.2	318.7
MaskFormer [6]	38.2	65.1	37.9	44.6	71.9	45.8	45.0	174.2
Mask2Former [5]	39.4	67.7	38.5	45.8	73.6	47.5	43.9	241.0
OSFormer [33]	41.0	**71.1**	40.8	42.5	72.5	42.3	46.6	324.7
DCNet (ours)	**45.3**	70.7	**47.5**	**52.8**	**77.1**	**56.5**	53.4	207.0

➢ Comparison of different components in PCD

	COD10K	NC4K
PCD (ours)	**45.3**	**52.8**
−camouflage attribute extractor	43.1 (−2.2)	48.1 (−4.7)
−difference attention mechanism	44.1 (−1.2)	51.3 (−1.5)
−both 2 components above	40.2 (−5.1)	46.4 (−6.4)

➢ Comparison of different attention mechanisms in ICS

Attention Mechanism	COD10K	NC4K
cross-attention [39]	42.4	49.5
masked attention [5]	44.7	51.7
reference attention (ours)	**45.3**	**52.8**

DSEL DEEP SPACE EXPLORATION LAB 深空探测实验室

Camouflaged Instance Segmentation via Explicit De-camouflaging

JUNE 18-22, 2023 CVPR VANCOUVER, CANADA

Qualitative Results

- Code will be released at: https://github.com/USTCL/DCNet