Visual Intelligence

REVIEW Open Access

Advances in deep concealed scene understanding

Deng-Ping Fan[1*], Ge-Peng Ji[2], Peng Xu[3], Ming-Ming Cheng[4], Christos Sakaridis[1] and Luc Van Gool[1]

Abstract

Concealed scene understanding (CSU) is a hot computer vision topic aiming to perceive objects exhibiting camouflage. The current boom in terms of techniques and applications warrants an up-to-date survey. This can help researchers better understand the global CSU field, including both current achievements and remaining challenges. This paper makes four contributions: (1) For the first time, we present a comprehensive survey of deep learning techniques aimed at CSU, including a taxonomy, task-specific challenges, and ongoing developments. (2) To allow for an authoritative quantification of the state-of-the-art, we offer the largest and latest benchmark for concealed object segmentation (COS). (3) To evaluate the generalizability of deep CSU in practical scenarios, we collected the largest concealed defect segmentation dataset termed CDS2K with the hard cases from diversified industrial scenarios, on which we constructed a comprehensive benchmark. (4) We discuss open problems and potential research directions for CSU.

Keywords: Concealed scene understanding, Segmentation, Detection, Survey, Introductory, Taxonomy, Deep learning, Machine learning

1 Introduction

Concealed scene understanding (CSU) aims to recognize objects that exhibit different forms of camouflage, as in Fig. 1. By its very nature, CSU is clearly a challenging problem compared with conventional object detection [1, 2]. It has numerous real-world applications, including search-and-rescue work, rare species discovery, healthcare (e.g., automatic diagnosis of colorectal polyps [3, 4] and lung lesions [5]), agriculture (e.g., pest identification [6] and fruit ripeness assessment [7]), and content creation (e.g., recreational art [8]). In the past decade, both academia and industry have widely studied CSU, and various types of images with camouflaged objects have been handled with traditional computer vision and pattern recognition techniques, including hand-engineered patterns (e.g., motion cues [9, 10] and optical flow [11, 12]), heuristic priors (e.g., color [13], texture [14] and intensity [15, 16]) and a combination of techniques [17–19].

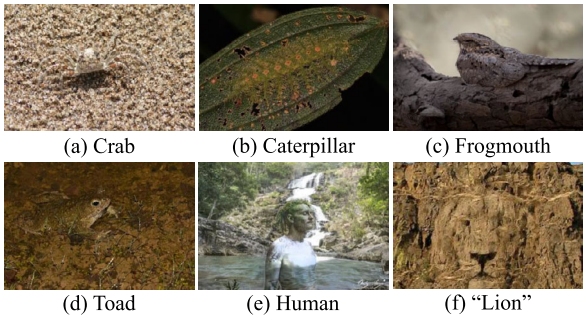

Figure 1 Sample gallery of concealment cases. (a)–(d) show images of animals in their natural habitat, selected from [20]. (e) depicts a concealed human in art from [21]. (f) features a synthesized "lion" by [22].

*Correspondence: dengpfan@gmail.com
[1]CVL, ETH Zurich, Zurich 8092, Switzerland
Full list of author information is available at the end of the article

In recent years, thanks to benchmarks becoming available (e.g., COD10K [20, 23] and NC4K [24]) and the rapid development of deep learning, this field has made important strides forward. In 2020, Fan et al. [20] released the first large-scale public dataset - COD10K - geared towards the advancement of perception tasks having to deal with concealment. This has also inspired other related disciplines. For instance, Mei et al. [25, 26] proposed a distraction-aware framework for the segmentation of camouflaged objects, which can be extended to the identification of transparent materials in natural scenes [27]. In 2023, Ji et al. [28] developed an efficient model that learns textures from object-level gradients, and its generalizability has been verified through diverse downstream applications, e.g., medical polyp segmentation and road crack detection.

Although multiple research teams have addressed tasks concerned with concealed objects, we believe that stronger interactions between the ongoing efforts would be beneficial. Thus, we mainly review the state and recent deep learning-based advances in CSU. Meanwhile, we contribute a large-scale concealed defect segmentation dataset termed CDS2K. This dataset consists of hard cases from diverse industrial scenarios, thus providing an effective benchmark for CSU.

Previous surveys and scope To the best of our knowledge, only a few survey papers have been published in the CSU community, which [29, 30] mainly reviewed non-deep learning techniques. There are some benchmarks [31, 32] with narrow scopes, such as image-level segmentation, where only a few deep learning methods were evaluated. In this paper, we present a comprehensive survey of deep learning CSU techniques, thus widening the scope. We also offer more extensive benchmarks with a more comprehensive comparison and with an application-oriented evaluation.

Contributions Our contributions are summarized as follows: (1) We represent the initial effort to thoroughly examine deep learning techniques tailored towards CSU thoroughly. This includes an overview of its classification and specific obstacles, as well as an assessment of its advancement during the era of deep learning, achieved through an examination of existing datasets and techniques. (2) To provide a quantitative evaluation of the current state-of-the-art, we have created a new benchmark for concealed object segmentation (COS), which is a crucial and highly successful area within CSU. It is the most up-to-date and comprehensive benchmark available. (3) To assess the applicability of CSU with deep learning in real-world scenarios, we have restructured the CDS2K dataset – the largest dataset for concealed defect segmentation – to include challenging cases from various industrial settings. We have utilized this updated dataset to create a comprehensive benchmark for evaluation. (4) Our discussion delves into the present obstacles, available prospects, and future research areas for the CSU community.

2 Background

2.1 Task taxonomy and formulation

2.1.1 Image-level CSU

In this section, we introduce five commonly used image-level CSU tasks, which can be formulated as a mapping function $\mathcal{F} : \mathbf{X} \mapsto \mathbf{Y}$ that converts the input space $\mathbf{X}$ into the target space $\mathbf{Y}$.

• *Concealed object segmentation (COS)* [23, 28] is a class-agnostic dense prediction task that segments concealed regions or objects with unknown categories. As presented in Fig. 2(a), the model $\mathcal{F}_{\text{COS}} : \mathbf{X} \mapsto \mathbf{Y}$ is supervised by a binary mask $\mathbf{Y}$ to predict a probability $\mathbf{p} \in [0, 1]$ for each pixel $\mathbf{x}$ of image $\mathbf{X}$, which is the confidence level that the model determines whether $\mathbf{x}$ belongs to the concealed region.

• *Concealed object localization (COL)* [24, 33] aims to identify the most noticeable region of concealed objects, which is in line with human perception psychology [33]. This task is to learn a dense mapping function $\mathcal{F}_{\text{COL}} : \mathbf{X} \mapsto \mathbf{Y}$. The output $\mathbf{Y}$ is a non-binary fixation map captured by an eye tracker device, as illustrated in Fig. 2(b). Essentially, the probability prediction $\mathbf{p} \in [0, 1]$ for a pixel $\mathbf{x}$ indicates how conspicuous its camouflage is.

• *Concealed instance ranking (CIR)* [24, 33] ranks different instances in a concealed scene based on their detectability. The level of camouflage is used as the basis for this ranking. The objective of the CIR task is to learn a dense mapping $\mathcal{F}_{\text{CIR}} : \mathbf{X} \mapsto \mathbf{Y}$ between the input space $\mathbf{X}$ and the camouflage ranking space $\mathbf{Y}$, where $\mathbf{Y}$ represents per-pixel annotations for each instance with corresponding rank levels. For example, in Fig. 2(c), there are three toads with different camouflage levels, and their ranking labels are from [24]. To perform this task, one can replace the category ID for each instance with rank labels in instance segmentation models such as Mask R-CNN [34].

• *Concealed instance segmentation (CIS)* [35, 36] is a technique that aims to identify instances in concealed scenarios based on their semantic characteristics. Unlike general instance segmentation [37, 38], where each instance is assigned a category label, CIS recognizes the attributes of concealed objects to distinguish between different entities more effectively. To achieve this objective, CIS employs a mapping function $\mathcal{F}_{\text{CIS}} : \mathbf{X} \mapsto \mathbf{Y}$, where $\mathbf{Y}$ is a scalar set comprising various entities used to parse each pixel. This concept is illustrated in Fig. 2(d).

• *Concealed object counting (COC)* [39] is a newly emerging topic in CSU that aims to estimate the number of instances concealed within their surroundings. As illustrated in Fig. 2(e), the COC estimates the center coordinates for each instance and generates their counts. Its formulation

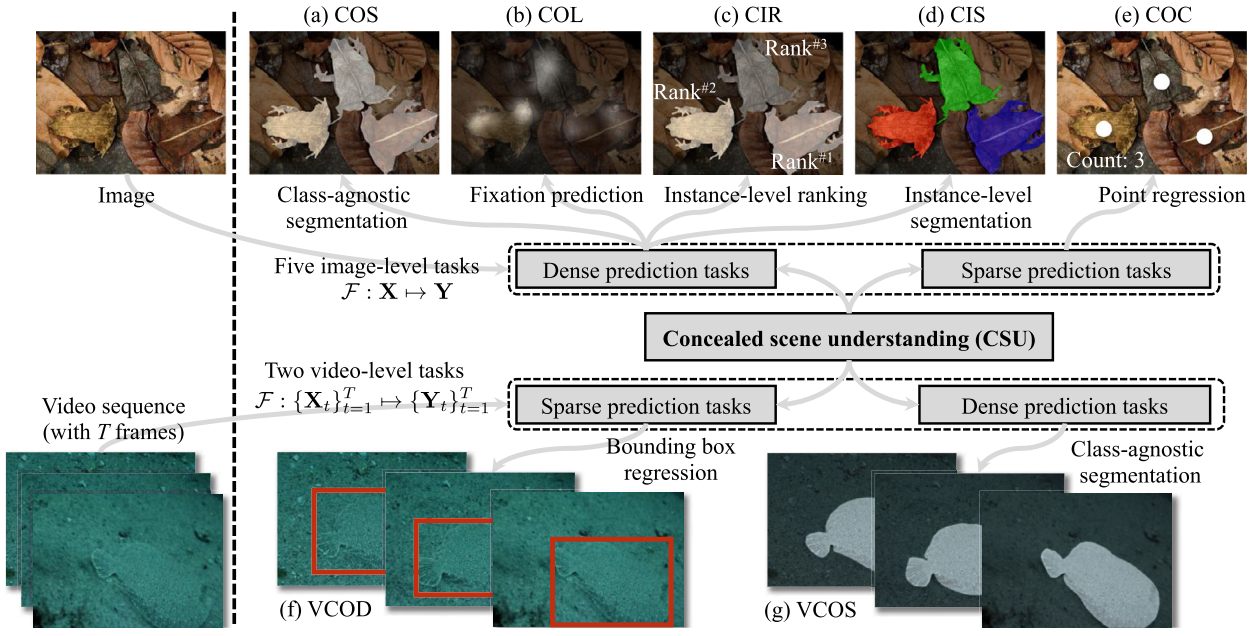

Figure 2 Illustration of the representative CSU tasks. Five of these are image-level tasks: (a) concealed object segmentation (COS), (b) concealed object localization (COL), (c) concealed instance ranking (CIR), (d) concealed instance segmentation (CIS), and (e) concealed object counting (COC). The remaining two are video-level tasks: (f) video concealed object detection (VCOD) and (g) video concealed object segmentation (VCOS). Each task has its own corresponding annotation visualization, which is explained in detail in Section 2.1

can be defined as $\mathcal{F}_{\text{COC}}: \mathbf{X} \mapsto \mathbf{Y}$, where $\mathbf{X}$ is the input image and $\mathbf{Y}$ represents the output density map that indicates the concealed instances in scenes.

Overall, the image-level CSU tasks can be categorized into two groups based on their semantics: object-level (COS and COL) and instance-level (CIR, COC, and CIS). Object-level tasks focus on perceiving objects while instance-level tasks aim to recognize semantics to distinguish different entities. Additionally, COC is regarded as a sparse prediction task based on its output form, whereas the others belong to dense prediction tasks. Among the literature reviewed in Table 1, COS has been extensively studied while research on the other three tasks is gradually increasing.

2.1.2 *Video-level CSU*

Given a video clip $\{\mathbf{X}_t\}_{t=1}^T$ containing T concealed frames, the video-level CSU can be formulated as a mapping function $\mathcal{F}: \{\mathbf{X}_t\}_{t=1}^T \mapsto \{\mathbf{Y}_t\}_{t=1}^T$ for parsing dense spatial-temporal correspondences, where $\mathbf{Y}_t$ is the label of frame $\mathbf{X}_t$.

• *Video concealed object detection (VCOD)* [40] is similar to video object detection [41]. This task aims to identify and locate concealed objects within a video by learning a spatial-temporal mapping function $\mathcal{F}_{\text{VCOD}}: \{\mathbf{X}_t\}_{t=1}^T \mapsto \{\mathbf{Y}_t\}_{t=1}^T$ that predicts the location $\mathbf{Y}_t$ of an object for each frame $\mathbf{X}_t$. The location label $\mathbf{Y}_t$ is provided as a bounding box (see Fig. 2(f)) consisting of four numbers (x, y, w, h) indicating the target's location. Here, (x, y) represents its top-left coordinate, while w and h denote its width and height, respectively.

• *Video concealed object segmentation (VCOS)* [42] originated from the task of camouflaged object discovery [40]. Its goal is to segment concealed objects within a video. This task usually utilizes spatial-temporal cues to drive the models to learn the mapping $\mathcal{F}_{\text{VCOS}}: \{\mathbf{X}_t\}_{t=1}^T \mapsto \{\mathbf{Y}_t\}_{t=1}^T$ between input frames $\mathbf{X}_t$ and corresponding segmentation mask labels $\mathbf{Y}_t$. Figure 2(g) shows an example of its segmentation mask.

In general, compared to image-level CSU, video-level CSU is developing relatively slowly, because collecting and annotating video data is labor-intensive and time-consuming. However, with the establishment of the first large-scale VCOS benchmark on MoCA-Mask [42], this field has made fundamental progress while still having ample room for exploration.

2.1.3 *Task relationship*

Among image-level CSU tasks, the CIR task requires the highest level of understanding as it may not only involve four subtasks, e.g., segmenting pixel-level regions (i.e., COS), counting (i.e., COC), or distinguishing different instances (i.e., CIS), but also ranking these instances according to their fixation probabilities (i.e., COL) under different difficulty levels. Additionally, regarding two video-level

Table 1 Essential characteristics of reviewed image-based methods. This summary outlines the key characteristics, including: *Architecture Design (Arc.)*: The framework used, which can be multi-stream (MSF), bottom-up & top-down (BTF), or branched (BF) frameworks. *Multiple Cues (M.C.)*: Whether an auxiliary cue is supplied. *Supervision Level (S.L.)*: Whether fully-supervised (★) or weakly-supervised (◇) learning is used. *Task Level (T.L.)*: The specific tasks addressed by the method, including COS (●), CIS (○), COC (□), and multi-task learning (■). N/A indicates that the source code is not available. For more detailed descriptions of these characteristics, please refer to Section 3.1 on image-level CSU models

#	Model	Pub.	Core component	Arc.	M.C.	S.L.	T.L.	Code
1	ANet [21]	$CVIU_{19}$	Classification & segmentation streams	BF	✓	★	●	https://sites.google.com/view/ltnghia/research/camo
2	SINet [20]	$CVPR_{20}$	Search and identification modules	BTF	-	★	●	https://github.com/DengPingFan/SINet
3	MirrorNet [110]	$Access_{21}$	Fuse input and mirror data streams	MSF	-	★	●	https://sites.google.com/view/ltnghia/research/camo
4	DCE [111]	$arXiv_{21}$	Depth contribution exploration, confidence-aware loss	BF	✓	★	●	https://github.com/JingZhang617/RGBD-COD
5	D2CNet [112]	TIE_{21}	Dual-branch, dual-guidance & cross-refine	BTF	-	★	●	https://github.com/MS-KangWang/COD-D2Net
6	C2FNet [113]	$IJCAI_{21}$	Context-aware cross-level fusion	BTF	-	★	●	https://github.com/thograce/C2FNet
7	UR-COD [114]	$MMAsia_{21}$	Uncertainty of pseudo-edge labels	MSF	-	★	●	https://github.com/nobukatsu-kajiura/UR-COD
8	TINet [115]	$AAAI_{21}$	Texture perception & feature interaction guidance	BF	✓	★	●	N/A
9	JSCOD [108]	$CVPR_{21}$	Uncertainty-aware adversarial learning	MSF	-	★	●	https://github.com/JingZhang617/Joint_COD_SOD
10	LSR [24]	$CVPR_{21}$	Localizing, segmenting, & ranking objects simultaneously	BF	✓	★	■	https://github.com/JingZhang617/COD-Rank-Localize-and-Segment
11	MGL [116]	$CVPR_{21}$	Mutual graph learning	BF	✓	★	●	https://github.com/fanyang587/MGL
12	PFNet [25]	$CVPR_{21}$	Distraction mining, positioning and focus modules	BTF	-	★	●	https://mhaiyang.github.io/CVPR2021_PFNet/index
13	UGTR [117]	$ICCV_{21}$	Uncertainty-guided transformer reasoning	BF	✓	★	●	https://github.com/fanyang587/UGTR
14	BAS [118]	$arXiv_{22}$	Residual refinement module, hybrid loss	BTF	-	★	●	https://github.com/xuebinqin/BASNet
15	OSFormer [35]	$ECCV_{22}$	Location-sensing transformer, coarse-to-fine fusion	BF	✓	★	○	https://github.com/PJLallen/OSFormer
16	CFL [36]	TIP_{22}	Camouflage fusion learning	BF	✓	★	○	https://sites.google.com/view/ltnghia/research/camo_plus_plus
17	NCHIT [119]	$CVIU_{22}$	Neighbor connection, hierarchical information transfer	BTF	-	★	●	N/A
18	DTC-Net [120]	TMM_{22}	Local bilinear & spatial coherence organization	BTF	-	★	●	N/A
19	C2FNet-V2 [121]	$TCSVT_{22}$	Context-aware cross-level fusion	BTF	-	★	●	https://github.com/Ben57882/C2FNet-TSCVT
20	CubeNet [122]	PR_{22}	Encoder-decoder framework with $\mathcal{X}$-connection	BF	✓	★	●	https://github.com/mczhuge/CubeNet
21	ERRNet [123]	PR_{22}	Selective edge aggregation, reversible re-calibration	BF	✓	★	●	https://github.com/GewelsJI/ERRNet
22	TPRNet [124]	$TVCJ_{22}$	Transformer-induced progressive refinement	BTF	-	★	●	https://github.com/zhangqiao970914/TPRNet
23	ANSA-Net [125]	$IJCNN_{22}$	Attention-based neighbor selective aggregation	BF	✓	★	●	N/A
24	BSANet [126]	$AAAI_{22}$	Boundary-guided separated attention	BF	✓	★	●	https://github.com/zhuhongwei1999/BSA-Net
25	FAPNet [127]	TIP_{22}	Boundary guidance, feature aggregation & propagation	BF	✓	★	●	https://github.com/taozh2017/FAPNet
26	FindNet [128]	TIP_{22}	Boundary-and-texture cues (extension of [126])	BF	✓	★	●	N/A
27	PINet [129]	$ICME_{22}$	Cascaded decamouflage module, label reweighting	BTF	-	★	●	N/A
28	OCENet [130]	$WACV_{22}$	Online confidence estimation, dynamic uncertainty loss	BF	✓	★	●	https://github.com/Carlisle-Liu/OCENet
29	BGNet [131]	$IJCAI_{22}$	Edge-guidance feature & context aggregation modules	BF	✓	★	●	https://github.com/thograce/BGNet
30	PreyNet [132]	MM_{22}	Bidirectional bridging interaction, predator learning	BF	✓	★	●	https://github.com/sxu1997/PreyNet
31	DTINet [133]	$ICPR_{22}$	Dual-task interactive transformer	BF	✓	★	●	https://github.com/liuzywen/COD
32	ZoomNet [134]	$CVPR_{22}$	Scale integration & hierarchical mixed-scale units	MSF	-	★	●	https://github.com/lartpang/ZoomNet
33	FDNet [135]	$CVPR_{22}$	Frequency enhancement & high-order relation modules	MSF	-	★	●	N/A
34	SegMaR [136]	$CVPR_{22}$	Segmenting, magnifying, reiterating in a iterative manner	BTF	-	★	●	https://github.com/dlut-dimt/SegMaR
35	SINetV2 [23]	$TPAMI_{22}$	Neighbor connection decoder, group-reversal attention	BTF	-	★	●	https://github.com/GewelsJI/SINet-V2
36	MGL-V2 [137]	TIP_{23}	Multi-source attention recovery (extension of [116])	BF	✓	★	●	https://github.com/fanyang587/MGL
37	FBNet [138]	$TMCCA_{23}$	Frequency-aware context aggregation & attention	BTF	-	★	●	N/A
38	TANet [139]	$TCSVT_{23}$	Texture-aware refinement, boundary-consistency loss	BTF	-	★	●	N/A
39	LSR+ [33]	$TCSVT_{23}$	Triple task learning (extension of [24])	BF	✓	★	■	https://github.com/JingZhang617/COD-Rank-Localize-and-Segment
40	SARNet [140]	$TCSVT_{23}$	Triple-stage refinement (search-amplify-recognize)	BTF	-	★	●	https://github.com/Haozhe-Xing/SARNet
41	MFFN [141]	$WACV_{23}$	Co-attention of multi-view, channel fusion unit	MSF	-	★	●	https://github.com/dwardzheng/MFFN_COD
42	CRNet [142]	$AAAI_{23}$	Feature-guided and consistency losses	MSF	-	◇	●	https://github.com/dddraxxx/Weakly-Supervised-Camouflaged-Object-Detection-with-Scribble-Annotations
43	HitNet [143]	$AAAI_{23}$	High-resolution iterative feedback	BTF	-	★	●	https://github.com/HUuxiaobin/HitNet
44	DGNet [28]	MIR_{23}	Gradient-based texture learning, efficient network	BF	✓	★	●	https://github.com/GewelsJI/DGNet
45	FSPNet [144]	$CVPR_{23}$	Feature shrinkage pyramid with transformer	BTF	-	★	●	https://github.com/ZhouHuang23/FSPNet
46	FEDER [145]	$CVPR_{23}$	Deep wavelet-like decomposition	BTF	-	★	●	https://github.com/ChunmingHe/FEDER
47	DCNet [146]	$CVPR_{23}$	Pixel-level decoupling, instance-level suppression	BF	✓	★	○	https://github.com/USTCL/DCNet
48	IOCFormer [39]	$CVPR_{23}$	Unify density- and regression-based strategies	BF	✓	★	□	https://github.com/GuoleiSun/Indiscernible-Object-Counting
49	PFNet+ [26]	$SCIS_{23}$	Extension of PFNet [25]	BTF	-	★	●	https://github.com/Mhaiyang/PFNet_Plus
50	DQnet [147]	$arXiv_{23}$	Cross-model detail querying, relation-based querying	MSF	-	★	●	https://github.com/CVPR23/DQnet
51	CamoFormer [148]	$arXiv_{23}$	Masked separable attention	BTF	-	★	●	https://github.com/HVision-NKU/CamoFormer
52	PopNet [149]	$arXiv_{23}$	Source-free depth, object pop-out prior	MSF	-	★	●	https://github.com/Zongwei97/PopNet

tasks, VCOS is a downstream task for VCOD since the segmentation task requires the model to provide pixel-level classification probabilities.

2.2 Related topics

Next, we will briefly introduce salient object detection (SOD), which, like COS, requires extracting properties of target objects, but one focuses on saliency while the other on the concealed attribute.

• *Image-level SOD* aims to identify the most attractive objects in an image and extract their pixel-accurate silhouettes [43]. Various network architectures have been explored in deep SOD models, e.g., multi-layer perceptron [44–47], fully convolutional [48–52], capsule-based [53–55], transformer-based [56], and hybrid [57, 58] networks. Meanwhile, different learning strategies are also studied in SOD models, including data-efficient methods (e.g., weakly-supervised with categorical tags [59–63] and unsupervised with pseudo masks [64–66]), multi-task paradigms (e.g., object subitizing [67, 68], fixation prediction [69, 70], semantic segmentation [71, 72], edge detection [73–77], image captioning [78]), and instance-level paradigms [79–82]. To learn more about this field comprehensively, readers can refer to popular surveys or representative studies on visual attention [83], saliency prediction [84], co-saliency detection [85–87], RGB SOD [1, 88–90], RGB-D (depth) SOD [91, 92], RGB-T (thermal) SOD [93, 94], and light-field SOD [95].

• *Video-level SOD.* The early development of video salient object detection (VSOD) originated from introducing attention mechanisms in video object segmentation (VOS) tasks. At that stage, the task scenes were relatively simple, containing only one object moving in the video. As moving objects tend to attract visual attention, VOS and VSOD were equivalent tasks. For instance, Wang et al. [96] used a fully convolutional neural network to address the VSOD task. With the development of VOS techniques, researchers introduced more complex scenes (e.g., with complex backgrounds, object movements, and two objects), but the two tasks remained equivalent. Thus, later works have exploited semantic-level spatial-temporal features [97–100], recurrent neural networks [101, 102], or offline motion cues such as optical flow [101, 103–105]. However, with the introduction of more challenging video scenes (containing three or more objects, simultaneous camera, and object movements), VOS and VSOD were no longer equivalent. However, researchers continued to approach the two tasks as equivalent, ignoring the issue of visual attention allocation in multi-object movement in video scenes, which seriously hindered the development of the field. To address this issue, in 2019, Fan et al. [106] introduced eye trackers to mark the changes in visual attention in multi-object movement scenarios, for the first time posing the scientific problem of *attention shift* in VSOD tasks, and constructed the first large-scale VSOD benchmark – DAVSOD,[1] as well as the baseline model SSAV, which propelled VSOD into a new stage of development.

• *Remarks.* COS and SOD are distinct tasks, but they can mutually benefit via the CamDiff approach [107]. This has been demonstrated through adversarial learning [108], leading to joint research efforts such as the recently proposed dichotomous image segmentation [109]. In Section 6, we will explore potential directions for future research in these areas.

3 CSU models with deep learning

This section systematically reviews CSU with deep learning approaches based on task definitions and data types. We have also created a GitHub base[2] as a comprehensive collection to provide the latest information in this field.

3.1 Image-level CSU models

We review the existing four image-level CSU tasks: concealed object segmentation (COS), concealed object localization (COL), concealed instance ranking (CIR), and concealed instance segmentation (CIS). Table 1 summarizes the key features of these reviewed approaches.

3.1.1 Concealed object segmentation

This section discusses previous solutions for camouflage object segmentation (COS) from two perspectives: network architecture and learning paradigm.

• *Network architecture.* Generally, fully convolutional networks (FCNs [150]) are the standard solution for image segmentation because they can receive the input of a flexible size and undergo a single feed-forward propagation. As expected, FCN-shaped frameworks dominate the primary solutions for COS, which fall into three categories:

(1) *Multi-stream framework*, shown in Fig. 3(a), contains multiple input streams to learn multi-source representations explicitly. MirrorNet [110] was the first attempt to add an extra data stream as a bio-inspired attack, which can break the camouflaged state. Several recent works have adopted a multi-stream approach to improve their results, such as supplying pseudo-depth generation [149], pseudo-edge uncertainty [114], adversarial learning paradigm [108], frequency enhancement stream [135], multi-scale [134] or multi-view [141] inputs, and multiple backbones [147]. Unlike other supervised settings, CRNet [142] is the only weakly-supervised framework that uses scribble labels as supervision. This approach helps alleviate overfitting problems on limited annotated data.

[1] https://github.com/DengPingFan/DAVSOD.

[2] https://github.com/GewelsJI/SINet-V2/blob/main/AWESOME_COD_LIST.md.

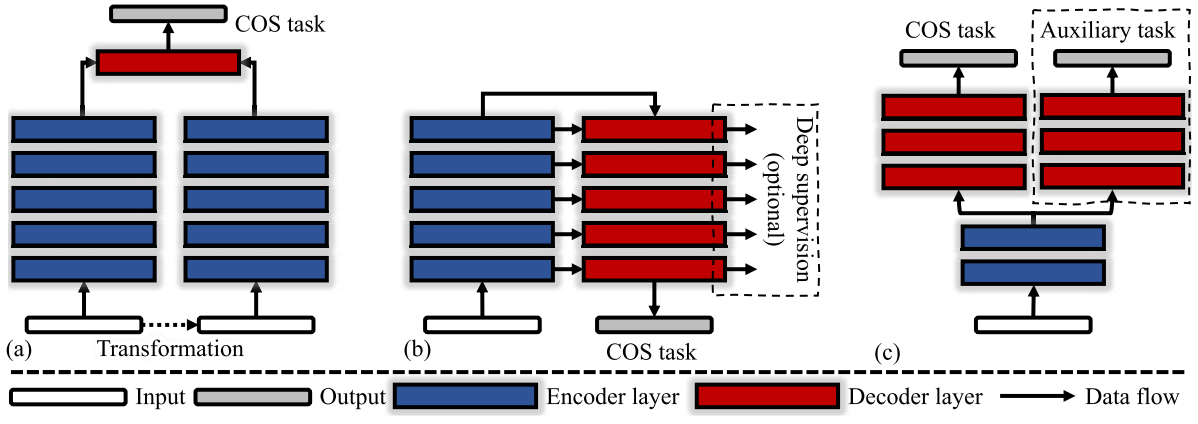

Figure 3 Network architectures for COS at a glance. We present four types of frameworks from left to right: (a) multi-stream framework, (b) bottom-up/top-down framework and its variant with deep supervision (optional), and (c) branched framework. See Section 3.1.1 for more details

(2) *Bottom-up and top-down framework*, as shown in Fig. 3(b), uses deeper features to enhance shallower features gradually in a single feed-forward pass. For example, C2FNet [113] adopts this design to improve concealed features from coarse-to-fine levels. In addition, SegMaR [136] employs an iterative refinement network with a sub-network based on this strategy. Furthermore, other studies [20, 23, 25, 26, 112, 118–121, 124, 125, 129, 138–140, 143–145, 148] utilized a deeply-supervised strategy [151, 152] on various intermediate feature hierarchies using this framework. This practice, also utilized by the feature pyramid network [153], combines more comprehensive multi-context features through dense top-down and bottom-up propagation and introduces additional supervision signals before final prediction to provide more dependable guidance for deeper layers.

(3) *Branched framework*, shown in Fig. 3(c), is a single-input-multiple-output architecture, consisting of both segmentation and auxiliary task branches. It should be noted that the segmentation part of this branched framework may have some overlap with previous frameworks, such as single-stream [21] and bottom-up and top-down [24, 28, 33, 108, 111, 115–117, 122, 123, 125–128, 130–133, 137] frameworks. For instance, ERRNet [123] and FAPNet [127] are typical examples of jointly learning concealed objects and their boundaries. Since these branched frameworks are closely related to the multi-task learning paradigm, we will provide further details.

• *Learning paradigm.* We discuss two common types of learning paradigms for COS tasks: single-task and multi-task.

(1) *Single-task learning* is the most commonly used paradigm in COS, which involves only a segmentation task for concealed targets. Based on this paradigm, most current works [20, 23, 121] focus on developing attention modules to identify target regions.

(2) *Multi-task learning* introduces an auxiliary task to coordinate or complement the segmentation task, leading to robust COS learning. These multi-task frameworks can be implemented by conducting confidence estimation [108, 117, 130, 132], localization/ranking [24, 33], category prediction [21] tasks and learning depth [111, 149], boundary [116, 122, 123, 126, 127, 131], and texture [28, 115] cues of camouflaged objects.

3.1.2 Concealed instance ranking

There has been limited research conducted on this topic. Lv et al. [24] observed for the first time that existing COS approaches could not quantify the difficulty level of camouflage. Regarding this issue, they used an eye tracker to create a new dataset, called CAM-LDR [33], that contains instance segmentation masks, fixation labels, and ranking labels. They also proposed two unified frameworks, LSR [24] and its extension LSR+ [33], to simultaneously learn triple tasks, i.e., localizing, segmenting, and ranking camouflaged objects. The insight behind it is that discriminative localization regions could guide the segmentation of the full scope of camouflaged objects, and then, the detectability of different camouflaged objects could be inferred by the ranking task.

3.1.3 Concealed instance segmentation

This task advances the COS task from the regional to the instance level, a relatively new field compared with the COS. Then, Le et al. [36] built a new CIS benchmark, CAMO++, by extending the previous CAMO [21] dataset. They also proposed a camouflage fusion learning strategy to fine-tune existing instance segmentation models (e.g., Mask R-CNN [34]) by learning image contexts. Based on instance benchmarks such as in COD10K [20] and NC4K [24], the first one-stage transformer frame-

work, OSFormer [35], was proposed for this field by introducing two core designs: a location-sensing transformer and coarse-to-fine fusion. Recently, Luo et al. [146] proposed segmenting camouflaged instances with two designs: a pixel-level camouflage decoupling module and an instance-level camouflage suppression module.

3.1.4 Concealed object counting

Sun et al. [39] recently introduced a new challenge for the community called indiscernible object counting (IOC), which involves counting objects that are difficult to distinguish from their surroundings. They created IOCfish5K, a large-scale dataset containing high-resolution images of underwater scenes with many indiscernible objects (focusing on fish) and dense annotations to address the lack of appropriate datasets for this challenge. They also proposed a baseline model called IOCFormer by integrating density-based and regression-based methods in a unified framework.

Based on the above summaries, the COS task is experiencing a rapid development period, resulting in numerous contemporary publications each year. However, very few proposed solutions have been proposed for the COL, CIR, and CIS tasks. This suggests that these fields remain under-explored and offer significant room for further research. Notably, many previous studies are available as references (such as saliency prediction [84], salient object subitizing [68], and salient instance segmentation [82]), providing a solid foundation for understanding these tasks from a camouflaged perspective.

3.2 Video-level CSU models

There are two schools of thought for the video-level CSU task, including detecting and segmenting camouflaged objects from videos. Refer to Table 2 for details.

3.2.1 Video concealed object detection

Most works [156, 158] formulated this topic as the degradation problem of the segmentation task due to the scarcity of pixel-wise annotations. They, as usual, were trained on segmentation datasets (e.g., DAVIS [161] and FBMS [162]) but evaluated the generalizability performance on a video camouflaged object detection dataset, MoCA [40]. These methods consistently opt to extract offline optical flow as motion guidance for the segmentation task, but they diversify over the learning strategies, such as fully-supervised learning with real [40, 157, 160] or synthetic [155, 158] data and self-supervised learning [156, 159].

3.2.2 Video concealed object segmentation

Xie et al. [154] proposed the first work on camouflaged object discovery in videos. They used a pixel-trajectory recurrent neural network to cluster foreground motion for segmentation. However, this work is limited to a small-scale dataset, CAD [163]. Recently, based upon the localization-level dataset MoCA [40] with bounding box labels, Cheng et al. [42] extended this field by creating a large-scale VCOS benchmark MoCA-Mask with pixel-level masks. They also introduced a two-stage baseline SLTNet to implicitly utilize motion information.

From what we have reviewed above, the current approaches for VCOS tasks are still in a nascent state of development. Several concurrent works in well-established video segmentation fields (e.g., self-supervised correspondence learning [164–168], unified framework for different

Table 2 Essential characteristics of reviewed video-level methods. *Optical flow (O.F.):* whether pre-generating optical flow map. *Supervision level (S.L.):* full-supervision with real data (★) or synthetic data (♣), and self-supervision (♡). *Task level (T.L.):* video camouflaged object detection (△) and segmentation (▲). For further details, refer to Section 3.2

#	Model	Pub.	Core components	O.F.	S.L.	T.L.	Project
1	FMC [154]	$CVPR_{19}$	Pixel trajectory recurrent neural network and clustering	✓	★	▲	N/A
2	VRS [40]	$ACCV_{20}$	Video registration and motion segmentation network	✓	★	△	https://github.com/hlamdouar/MoCA/
3	SIMO [155]	$BMVC_{21}$	Dual-head architecture, synthetic dataset	✓	♣	△	https://www.robots.ox.ac.uk/~vgg/research/simo/
4	MG [156]	$ICCV_{21}$	Self-supervised motion grouping	✓	♡	△	https://github.com/charigyang/motiongrouping
5	RCF [157]	$arXiv_{22}$	Rotation-compensated flow, camera motion estimation	✓	★	△	N/A
6	OCLR [158]	$NeurIPS_{22}$	Object-centric layered representation, synthetic dataset	✓	♣	△	N/A
7	OFS [159]	$TPAMI_{22}$	Expectation-maximization method, motion augmentation	✓	♡	△	https://github.com/Etienne-Meunier-Inria/EM-Flow-Segmentation
8	QSDI [160]	$CVPR_{22}$	Quantifying the static and dynamic biases	✓	★	△	https://yorkucvil.github.io/Static-Dynamic-Interpretability/
9	SLTNet [42]	$CVPR_{22}$	Implicit motion handling, short- and long-term modules	-	★	▲	https://github.com/XuelianCheng/SLT-Net

motion-based tasks [169–171]) points the way to further exploration. In addition, considering high-level semantic understanding has a research gap that merits being supplied, such as semantic segmentation and instance segmentation in the camouflaged scenes.

4 CSU datasets

In recent years, various datasets have been collected for both image- and video-level CSU tasks. In Table 3, we summarize the features of the representative datasets.

4.1 Image-level datasets

• *CAMO-COCO* [21] is tailor-made for COS tasks with 2500 image samples across eight categories, divided into two sub-datasets, i.e., CAMO with camouflaged objects and MS-COCO with non-camouflaged objects. Both CAMO and MS-COCO contain 1250 images with a split of 1000 for training and 250 for testing.

• *NC4K* [24] is currently the largest testing set for evaluating COS models. It consists of 4121 camouflaged images sourced from the Internet and can be divided into two primary categories: natural scenes and artificial scenes. In addition to the images, this dataset also provides localization labels that include both object-level segmentation and instance-level masks, making it a valuable resource for researchers working in this field. In a recent study by Lv et al. [24], an eye tracker was utilized to collect fixation information for each image. As a result, a CAM-FR dataset of 2280 images was created, with 2000 images used for training and 280 for testing. The dataset was annotated with three types of labels: localization, ranking, and instance labels.

• *CAMO++* [36] is a newly released dataset that contains 5500 samples, all of which have undergone hierarchical pixel-wise annotation. The dataset is divided into two parts: camouflaged samples (1700 images for training and 1000 for testing) and non-camouflaged samples (1800 images for training and 1000 for testing).

• *COD10K* [20, 23] is currently the largest-scale dataset, featuring a wide range of camouflaged scenes. It contains 10,000 images from multiple open-access photography websites, covering 10 super-classes and 78 sub-classes. Of these images, 5066 are camouflaged, 1934 are non-camouflaged pictures, and 3000 are background images. The camouflaged subset of COD10K is annotated using different labels such as category labels, bounding boxes, object-level masks, and instance-level masks, providing a diverse set of annotations.

• *CAM-LDR* [33] comprises of 4040 training and 2026 testing samples. These samples were selected from commonly-used hybrid training datasets (i.e., CAMO with 1000 training samples and COD10K with 3040 training samples), along with the testing dataset (i.e., COD10K with 2026 testing samples). CAM-LDR is an extension of NC4K [24] that includes four types of annotations: localization labels, ranking labels, object-level segmentation masks, and instance-level segmentation masks. The ranking labels are categorized into six difficulty levels – background, easy, medium1, medium2, medium3, and hard.

• *S-COD* [142] is the first dataset designed specifically for the COS task under the weakly-supervised setting. The dataset includes 4040 training samples, with 3040 samples selected from COD10K and 1000 from CAMO. These samples were re-labeled using scribble annotations that provide a rough outline of the primary structure based on first impressions, without pixel-wise ground-truth information.

• *IOCfish5K* [39] is a distinct dataset that focuses on counting instances of fish in camouflaged scenes. This COC dataset comprises 5637 high-resolution images collected from YouTube, with 659,024 center points annotated. The dataset is divided into three subsets, with 3137 images allocated for training, 500 for validation, and 2000 for testing.

Remarks In summary, three datasets (CAMO, COD10K, and NC4K) are commonly used as benchmarks to evaluate camouflage object segmentation (COS) approaches, with the experimental protocols typically described in Section 5.2. For the concealed instance segmentation (CIS) task, two datasets (COD10K and NC4K) containing instance-level segmentation masks can be utilized. The CAM-LDR dataset, which provides fixation information and three types of annotations collected from a physical eye tracker device, is suitable for various brain-inspired explorations in computer vision. Additionally, there are two new datasets from CSU: S-COD, designed for weakly-supervised COS, and IOCfish5K, focused on counting objects within camouflaged scenes.

4.2 Video-level datasets

• *CAD* [163] is a small dataset comprising nine short video clips and 836 frames. The annotation strategy used in this dataset is sparse, with camouflaged objects being annotated every five frames. As a result, there are 191 segmentation masks available in the dataset.

• *MoCA* [40] is a comprehensive video database from YouTube that aims to detect moving camouflaged animals. It consists of 141 video clips featuring 67 categories and comprises 37,250 high-resolution frames with corresponding bounding box labels for 7617 instances.

• *MoCA-Mask* [42], an extension of the MoCA dataset [40], provides human-annotated segmentation masks every five frames based on the MoCA dataset [40]. MoCA-Mask is divided into two parts: a training set consisting of 71 short clips (19,313 frames with 3946 segmentation masks) and an evaluation set containing 16 short clips (3626 frames with 745 segmentation masks). To label those unlabeled frames, pseudo-segmentation labels were synthesized using a bidirectional optical flow-based strategy [172].

Table 3 Essential characteristics for CSU datasets. *Train/Test:* number of samples for training/testing (e.g., images for image dataset or frames for video dataset) *Task:* data type of dataset. *N. Cam.:* whether collecting non-camouflaged samples. *Cls.:* whether providing classification labels. *B. Box:* whether providing bounding box labels for the detection task. *Obj./Ins.:* whether providing object- or instance-level segmentation masks for segmentation tasks. *Rank:* whether providing ranking labels for instances. *Scr.:* whether providing weakly-supervised labels in scribbled form. *Cou.:* whether providing dense object counting labels. See Section 4.1 and Section 4.2 for more descriptions

#	Dataset	Year	Pub.	Train	Test	Task	N. Cam.	Cls.	B. Box	Obj.	Ins.	Fix.	Rank	Scr.	Cou.	Website
1	CAD [163]	2016	ECCV	0	836	Video	-	✓	-	✓	-	-	-	-	-	http://vis-www.cs.umass.edu/motionSegmentation/
2	CAMO-COCO [21]	2019	CVIU	2000	500	Image	✓	-	-	✓	-	-	-	-	-	https://sites.google.com/view/ltnghia/research/camo
3	MoCA [40]	2020	ACCV	0	37,250	Video	-	✓	✓	-	-	-	-	-	-	https://www.robots.ox.ac.uk/~vgg/data/MoCA/
4	NC4K [24]	2021	CVPR	0	4121	Image	-	-	✓	✓	✓	-	-	-	-	https://github.com/JingZhang617/COD-Rank-Localize-and-Segment
5	MoCA-Mask [42]	2022	CVPR	19,313	3626	Video	-	✓	-	✓	-	-	-	-	-	https://xueliancheng.github.io/SLT-Net-project/
6	CAMO++ [36]	2022	TIP	3500	2000	Image	✓	-	✓	✓	✓	-	-	-	-	https://sites.google.com/view/ltnghia/research/camo_plus_plus
7	COD10K [20, 23]	2022	TPAMI	6000	4000	Image	✓	✓	✓	✓	✓	-	-	-	-	https://dengpingfan.github.io/pages/COD.html
8	CAM-LDR [33]	2023	TCSVT	4040	2026	Image	-	-	-	✓	✓	✓	✓	-	-	https://github.com/JingZhang617/COD-Rank-Localize-and-Segment
9	S-COD [142]	2023	AAAI	4040	0	Image	-	-	-	-	-	-	-	✓	-	https://github.com/dddraxxx/Weakly-Supervised-Camouflaged-Object-Detection-with-Scribble-Annotations
10	IOCfish5K [39]	2023	CVPR	3637	2000	Image	-	✓	-	-	-	-	-	-	✓	https://github.com/GuoleiSun/Indiscernible-Object-Counting

Remarks The MoCA dataset is currently the largest collection of videos with concealed objects, while it only offers detection labels. As a result, researchers in the community [156, 158] typically assess the performance of well-trained segmentation models by converting segmentation masks into detection bounding boxes. Recently, there has been a shift towards video segmentation in concealed scenes with the introduction of MoCA-Mask. Despite these advancements, the quantity and quality of data annotations remain insufficient for constructing a reliable video model that can effectively handle complex concealed scenarios.

5 CSU benchmarks

In this investigation, our benchmarking is built on COS tasks since this topic is relatively well-established and offers a variety of competing approaches. The following sections will provide details over the evaluation metrics (Section 5.1), benchmarking protocols (Section 5.2), quantitative analyses (Section 5.3, Section 5.4, Section 5.5), and qualitative comparisons (Section 5.6).

5.1 Evaluation metrics

As suggested in [23], there are five commonly used metrics[3] available for COS evaluation. We compare a prediction mask $\mathbf{P}$ with its corresponding ground-truth mask $\mathbf{G}$ at the same image resolution.

• *MAE* (mean absolute error, $\mathcal{M}$) is a conventional pixel-wise measure, which is defined as:

$$\mathcal{M} = \frac{1}{W \times H} \sum_{x}^{W} \sum_{y}^{H} \left| \mathbf{P}(x,y) - \mathbf{G}(x,y) \right|, \tag{1}$$

where W and H are the width and height of $\mathbf{G}$ respectively, and (x, y) are pixel coordinates in $\mathbf{G}$.

• F-measure can be defined as:

$$F_\beta = \frac{(1+\beta^2)\text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}, \tag{2}$$

where $\beta^2 = 0.3$ is used to emphasize the precision value over the recall value, as recommended in [90]. Two other metrics are derived from:

$$\text{Precision} = \frac{|\mathbf{P}(T) \cap \mathbf{G}|}{|\mathbf{P}(T)|}, \qquad \text{Recall} = \frac{|\mathbf{P}(T) \cap \mathbf{G}|}{|\mathbf{G}|}, \tag{3}$$

where $\mathbf{P}(T)$ is a binary mask obtained by thresholding the non-binary predicted map $\mathbf{P}$ with a threshold value $T \in [0, 255]$. The symbol $|\cdot|$ calculates the total area of the mask inside the map. Therefore, it is possible to convert a non-binary prediction mask into a series of binary masks with threshold values ranging from 0 to 255. By iterating over all thresholds, three metrics are obtained with maximum (F_β^{mx}), mean (F_β^{mn}), and adaptive (F_β^{ad}) values of the F-measure.

• *Enhanced-alignment measure* (E_ϕ) [180, 181] is a recently proposed binary foreground evaluation metric, which considers both local and global similarity between two binary maps. Its formulation is defined as follows:

$$E_\phi = \frac{1}{W \times H} \sum_{x}^{W} \sum_{y}^{H} \phi\big[\mathbf{P}(x,y), \mathbf{G}(x,y)\big], \tag{4}$$

where ϕ is the enhanced-alignment matrix. Similar to F_β, this metric also includes three values computed over all the thresholds, i.e., maximum (E_ϕ^{mx}), mean (E_ϕ^{mn}), and adaptive (E_ϕ^{ad}) values.

• *Structure measure* ($\mathcal{S}_\alpha$) [182, 183] is used to measure the structural similarity between a non-binary prediction map and a ground-truth mask:

$$\mathcal{S}_\alpha = (1-\alpha)\mathcal{S}_o(\mathbf{P}, \mathbf{G}) + \alpha \mathcal{S}_r(\mathbf{P}, \mathbf{G}), \tag{5}$$

where α balances the object-aware similarity $\mathcal{S}_o$ and region-aware similarity $\mathcal{S}_r$. As in the original paper, we use the default setting for $\alpha = 0.5$.

5.2 Experimental protocols

As suggested by Fan et al. [23], all competing approaches in the benchmarking analysis were trained on a hybrid dataset comprising the training portions of the COD10K [20] and CAMO [21] datasets, totaling 4040 samples. The models were then evaluated on three popular used benchmarks: COD10K's testing portion with 2026 samples [20], CAMO with 250 samples [21], and NC4K with 4121 samples [24].

5.3 Quantitative analysis of CAMO

As reported in Table 4, we evaluated 36 deep learning-based approaches on the CAMO testing dataset [21] using various metrics. These models were classified into two groups based on the backbones they used: 32 convolutional-based and four transformer-based models. For those models using convolutional-based backbones, several interesting findings are observed and summarized as follows.

• CamoFormer-C [148] achieved the best performance on CAMO with the ConvNeXt [176] based backbone, even surpassing some metrics produced by transformer-based methods, such as $\mathcal{S}_\alpha$ value: 0.859 (CamoFormer-C) *vs.* 0.856 (DTINet [133]) *vs.* 0.849 (HitNet [143]). However, CamoFormer-R [148] with the ResNet-50 backbone was unable to outperform competitors with the same backbone, such as using multi-scale zooming (ZoomNet [134]) and iterative refinement (SegMaR [136]) strategies.

[3] https://github.com/DengPingFan/CSU/tree/main/cos_eval_toolbox.

Table 4 Quantitative comparison on the CAMO [21] testing set. We classify the competing approaches based on two aspects: those using convolution-based backbones such as ResNet [173], Res2Net [174], EffNet [175], and ConvNext [176]; and those using transformer-based backbones such as MiT [177], PVTv2 [178], and Swin [179]. We test two efficiency metrics, model parameters (Para) and multiply-accumulate operations (MACs), in accordance with the preset input resolution in the original paper. In addition, nine evaluation metrics are reported, and the best three scores are highlighted in **red**, **green**, and **blue**, respectively, with ↑/↓ indicating that higher/lower scores are better. If the results are unavailable since the code has not been made public, we use a hyphen (–) to denote it. We will follow these notations in subsequent tables unless otherwise specified

Model	Pub/Year	Backbone	Input	Para.	MACs	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$E_\phi^{ad} \uparrow$	$E_\phi^{mn} \uparrow$	$E_\phi^{mx} \uparrow$	$F_\beta^{ad} \uparrow$	$F_\beta^{mn} \uparrow$	$F_\beta^{mx} \uparrow$
• *Convolution-based Backbone*														
SINet [20]	$CVPR_{20}$	ResNet-50	352^2	48.95M	19.42G	0.745	0.644	0.092	0.825	0.804	0.829	0.712	0.702	0.708
D2CNet [112]	TIE_{21}	Res2Net-50	320^2	–	–	0.774	0.683	0.087	0.844	0.818	0.838	0.747	0.735	0.743
C2FNet [113]	$IJCAI_{21}$	Res2Net-50	352^2	28.41M	13.12G	0.796	0.719	0.080	0.865	0.854	0.864	0.764	0.762	0.771
TINet [115]	$AAAI_{21}$	ResNet-50	352^2	28.56M	8.58G	0.781	0.678	0.087	0.847	0.836	0.848	0.729	0.728	0.745
JSCOD [108]	$CVPR_{21}$	ResNet-50	352^2	121.63M	25.20G	0.800	0.728	0.073	0.872	0.859	0.873	0.779	0.772	0.779
LSR [24]	$CVPR_{21}$	ResNet-50	352^2	57.90M	25.21G	0.787	0.696	0.080	0.859	0.838	0.854	0.756	0.744	0.753
R-MGL [116]	$CVPR_{21}$	ResNet-50	473^2	67.64M	249.89G	0.775	0.673	0.088	0.848	0.812	0.842	0.738	0.726	0.740
S-MGL [116]	$CVPR_{21}$	ResNet-50	473^2	63.60M	236.60G	0.772	0.664	0.089	0.850	0.807	0.842	0.733	0.721	0.739
PFNet [25]	$CVPR_{21}$	ResNet-50	416^2	46.50M	26.54G	0.782	0.695	0.085	0.855	0.841	0.855	0.751	0.746	0.758
UGTR [117]	$ICCV_{21}$	ResNet-50	473^2	48.87M	127.12G	0.785	0.686	0.086	0.861	0.823	0.854	0.749	0.738	0.754
BAS [118]	$arXiv_{21}$	ResNet-34	288^2	87.06M	161.19G	0.749	0.646	0.096	0.808	0.796	0.808	0.696	0.692	0.703
NCHIT [119]	$CVIU_{22}$	ResNet-50	288^2	–	–	0.784	0.652	0.088	0.841	0.805	0.840	0.723	0.707	0.739
C2FNet-V2 [121]	$TCSVT_{22}$	Res2Net-50	352^2	44.94M	18.10G	0.799	0.730	0.077	0.869	0.859	0.869	0.777	0.770	0.779
CubeNet [122]	PR_{22}	ResNet-50	352^2	–	–	0.788	0.682	0.085	0.852	0.838	0.860	0.734	0.732	0.750
ERRNet [123]	PR_{22}	ResNet-50	352^2	69.76M	20.05G	0.779	0.679	0.085	0.855	0.842	0.858	0.731	0.729	0.742
TPRNet [124]	$TVCJ_{22}$	Res2Net-50	352^2	32.95M	12.98G	0.807	0.725	0.074	0.880	0.861	0.883	0.777	0.772	0.785
FAPNet [127]	TIP_{22}	Res2Net-50	352^2	29.52M	29.69G	0.815	0.734	0.076	0.877	0.865	0.880	0.776	0.776	0.792
BSANet [126]	$AAAI_{22}$	Res2Net-50	384^2	32.58M	29.70G	0.794	0.717	0.079	0.866	0.851	0.867	0.768	0.763	0.770
OCENet [130]	$WACV_{22}$	ResNet-50	480^2	60.31M	59.70G	0.802	0.723	0.080	0.866	0.852	0.865	0.776	0.766	0.777
BGNet [131]	$IJCAI_{22}$	Res2Net-50	416^2	79.85M	58.45G	0.812	0.749	0.073	0.876	0.870	0.882	0.786	0.789	0.799
PreyNet [132]	MM_{22}	ResNet-50	448^2	38.53M	58.10G	0.790	0.708	0.077	0.856	0.842	0.857	0.763	0.757	0.765
ZoomNet [134]	$CVPR_{22}$	ResNet-50	384^2	32.38M	95.50G	0.820	0.752	0.066	0.883	0.877	0.892	0.792	0.794	0.805
FDNet [135]	$CVPR_{22}$	Res2Net-50	416^2	–	–	**0.841**	**0.775**	**0.063**	**0.901**	**0.895**	0.908	**0.803**	**0.807**	**0.826**
SegMaR [136]	$CVPR_{22}$	ResNet-50	352^2	56.21M	33.63G	0.815	0.753	0.071	0.881	0.874	0.884	0.795	0.795	0.803
SINetV2 [23]	$TPAMI_{22}$	Res2Net-50	352^2	26.98M	12.28G	0.820	0.743	0.070	0.884	0.882	0.895	0.779	0.782	0.801
CamoFormer-C [148]	$arXiv_{23}$	ConvNeXt-B	384^2	96.69M	50.77G	**0.859**	**0.812**	**0.050**	**0.919**	**0.913**	**0.920**	**0.842**	**0.842**	**0.855**
CamoFormer-R [148]	$arXiv_{23}$	ResNet-50	384^2	54.25M	78.85G	0.816	0.712	0.076	0.863	0.874	**0.916**	0.735	0.745	0.813
PopNet [149]	$arXiv_{23}$	Res2Net-50	512^2	188.05M	154.88G	0.808	0.744	0.077	0.871	0.859	0.874	0.790	0.784	0.792
CRNet [142]	$AAAI_{23}$	ResNet-50	320^2	32.65M	11.83G	0.735	0.641	0.092	0.829	0.815	0.830	0.709	0.701	0.707
PFNet+ [26]	$SCIS_{23}$	ResNet-50	480^2	–	–	0.791	0.713	0.080	0.862	0.850	0.865	0.764	0.761	0.770
DGNet-S [28]	MIR_{23}	EffNet-B1	352^2	7.02M	2.77G	0.826	0.754	**0.063**	0.896	0.893	0.907	0.786	0.792	0.810
DGNet [28]	MIR_{23}	EffNet-B4	352^2	19.22M	1.20G	**0.839**	**0.769**	**0.057**	**0.906**	**0.901**	**0.915**	**0.804**	**0.806**	**0.822**
• *Transformer-based Backbone*														
DTINet [133]	$ICPR_{22}$	MiT-B5	256^2	266.33M	144.68G	**0.856**	0.796	**0.050**	**0.918**	**0.916**	**0.927**	0.821	0.823	**0.843**
CamoFormer-S [148]	$arXiv_{23}$	Swin-B	384^2	97.27M	64.13G	**0.876**	**0.832**	**0.043**	**0.935**	**0.930**	**0.938**	**0.856**	**0.856**	**0.871**
CamoFormer-P [148]	$arXiv_{23}$	PVTv2-B4	384^2	71.40M	39.74G	**0.872**	**0.831**	**0.046**	**0.931**	**0.929**	**0.938**	**0.853**	**0.854**	**0.868**
HitNet [143]	$AAAI_{23}$	PVTv2-B2	704^2	25.73M	55.95G	0.849	**0.809**	0.055	0.910	0.906	**0.910**	**0.833**	**0.831**	0.838

• For those Res2Net-based models, FDNet [135] achieved the top performance on CAMO with high-resolution input of 416^2. In addition, SINetV2 [23] and FAPNet [127] also achieved satisfactory results using the same backbone but with a small input size of 352^2.

• DGNet [28], is an efficient model that stands out with its top-3 performance compared to heavier models such as JSCOD [108] (121.63M) and PopNet [149] (181.05M), despite having only 19.22M parameters and 1.20G computation costs. Its performance-efficiency balance makes it a promising architecture for further exploration of its potential capabilities.

• Interestingly, CRNet [142] – a weakly-supervised model – competes favorably with the early fully-supervised model SINet [20]. This suggests that there is room for developing models to bridge the gap towards better data-efficient learning, e.g., self-/semi-supervised learning.

Furthermore, transformer-based methods can significantly improve performance due to their superior long-range modeling capabilities. Here, we test four transformer-based models on the CAMO testing dataset, yielding three noteworthy findings:

• CamoFormer-S [148], utilizes a Swin transformer design to enhance the hierarchical modeling ability on con-

Table 5 Quantitative comparison on the NC4K [24] testing dataset

Model	Pub/Year	Backbone	$\mathcal{S}_\alpha \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$E_\phi^{ad} \uparrow$	$E_\phi^{mn} \uparrow$	$E_\phi^{mx} \uparrow$	$F_\beta^{ad} \uparrow$	$F_\beta^{mn} \uparrow$	$F_\beta^{mx} \uparrow$
• *Convolution-based Backbone*											
SINet [20]	$CVPR_{20}$	ResNet-50	0.808	0.723	0.058	0.883	0.871	0.883	0.768	0.769	0.775
C2FNet [113]	$IJCAI_{21}$	Res2Net-50	0.838	0.762	0.049	0.901	0.897	0.904	0.788	0.795	0.810
TINet [115]	$AAAI_{21}$	ResNet-50	0.829	0.734	0.055	0.882	0.879	0.890	0.766	0.773	0.793
JSCOD [108]	$CVPR_{20}$	ResNet-50	0.842	0.771	0.047	0.906	0.898	0.907	0.803	0.806	0.816
LSR [24]	$CVPR_{21}$	ResNet-50	0.840	0.766	0.048	0.904	0.895	0.907	0.802	0.804	0.815
R-MGL [116]	$CVPR_{21}$	ResNet-50	0.833	0.740	0.052	0.890	0.867	0.893	0.778	0.782	0.800
S-MGL [116]	$CVPR_{21}$	ResNet-50	0.829	0.731	0.055	0.885	0.863	0.893	0.771	0.777	0.797
PFNet [25]	$CVPR_{21}$	ResNet-50	0.829	0.745	0.053	0.894	0.887	0.898	0.779	0.784	0.799
UGTR [117]	$ICCV_{21}$	ResNet-50	0.839	0.747	0.052	0.889	0.874	0.899	0.779	0.787	0.807
BAS [118]	$arXiv_{21}$	ResNet-34	0.817	0.732	0.058	0.868	0.859	0.872	0.767	0.772	0.782
NCHIT [119]	$CVIU_{22}$	ResNet-50	0.830	0.710	0.058	0.872	0.851	0.894	0.751	0.758	0.792
C2FNet-V2 [121]	$TCSVT_{22}$	Res2Net-50	0.840	0.770	0.048	0.900	0.896	0.904	0.799	0.802	0.814
ERRNet [123]	PR_{22}	ResNet-50	0.827	0.737	0.054	0.892	0.887	0.901	0.769	0.778	0.794
TPRNet [124]	$TVCJ_{22}$	Res2Net-50	0.846	0.768	0.048	0.901	0.898	0.911	0.798	0.805	0.820
FAPNet [127]	TIP_{22}	Res2Net-50	0.851	0.775	0.047	0.903	0.899	0.910	0.804	0.810	0.826
BSANet [126]	$AAAI_{22}$	Res2Net-50	0.841	0.771	0.048	0.906	0.897	0.907	0.805	0.808	0.817
OCENet [130]	$WACV_{22}$	ResNet-50	0.853	0.785	0.045	0.908	0.902	0.913	0.812	0.818	0.831
BGNet [131]	$IJCAI_{22}$	Res2Net-50	0.851	**0.788**	0.044	0.911	0.907	0.916	0.813	0.820	**0.833**
PreyNet [132]	MM_{22}	ResNet-50	0.834	0.763	0.050	0.899	0.887	0.899	0.805	0.803	0.811
ZoomNet [134]	$CVPR_{22}$	ResNet-50	0.853	0.784	**0.043**	0.907	0.896	0.912	0.814	0.818	0.828
FDNet [135]	$CVPR_{22}$	Res2Net-50	0.834	0.750	0.052	0.895	0.893	0.905	0.774	0.784	0.804
SegMaR [136]	$CVPR_{22}$	ResNet-50	0.841	0.781	0.046	0.905	0.896	0.907	**0.821**	**0.821**	0.826
SINetV2 [23]	$TPAMI_{22}$	Res2Net-50	0.847	0.770	0.048	0.901	0.903	0.914	0.792	0.805	0.823
CamoFormer-C [148]	$arXiv_{23}$	ConvNeXt-B	**0.883**	**0.834**	**0.032**	**0.937**	**0.933**	**0.940**	**0.851**	**0.857**	**0.870**
CamoFormer-R [148]	$arXiv_{23}$	ResNet-50	0.855	**0.788**	**0.042**	**0.913**	0.900	0.914	0.820	**0.821**	0.830
PopNet [149]	$arXiv_{23}$	Res2Net-50	**0.861**	**0.802**	**0.042**	**0.915**	**0.909**	**0.919**	**0.830**	**0.833**	**0.843**
DGNet-S [28]	MIR_{23}	EfficientNet-B1	0.845	0.764	0.047	0.902	0.902	0.913	0.789	0.799	0.819
DGNet [28]	MIR_{23}	EfficientNet-B4	**0.857**	0.784	**0.042**	0.910	**0.911**	**0.922**	0.803	0.814	**0.833**
• *Transformer-based Backbone*											
DTINet [133]	$ICPR_{22}$	MiT-B5	0.863	0.792	0.041	**0.914**	0.917	**0.926**	0.809	0.818	0.836
CamoFormer-S [148]	$arXiv_{23}$	Swin-B	**0.888**	**0.840**	**0.031**	**0.941**	**0.937**	**0.946**	**0.857**	**0.863**	**0.877**
CamoFormer-P [148]	$arXiv_{23}$	PVTv2-B4	**0.892**	**0.847**	**0.030**	**0.941**	**0.939**	**0.946**	**0.863**	**0.868**	**0.880**
HitNet [143]	$AAAI_{23}$	PVTv2-B2	**0.875**	**0.834**	**0.037**	**0.928**	**0.926**	**0.929**	**0.854**	**0.853**	**0.863**

cealed content, resulting in superior performance across the entire CAMO benchmark. We also observed that the PVT-based variant CamoFormer-P [148] achieved comparable results but with fewer parameters, i.e., 71.40M (CamoFormer-P) *vs.* 97.27M (CamoFormer-R).

• DTINet [133] is a dual-branch network that utilizes the MiT-B5 semantic segmentation model from SegFormer [177] as backbone. Despite having 266.33M parameters, it has not delivered impressive performance due to the challenges of balancing these two heavy branches. Nevertheless, this attempt defies our preconceptions and inspires us to investigate the generalizability of semantic segmentation models in concealed scenarios.

• We also investigated the impact of input resolution on the performance of different models. HitNet [143] uses a high-resolution image of 704^2, which can improve the detection of small targets, but at the expense of increased computation costs. Similarly, convolutional-based approaches such as ZoomNet [134] achieved impressive results by taking multiple inputs with different resolutions (the largest being 576^2) to enhance segmentation performance. However, not all models benefit from this approach. For instance, PopNet [149] with a resolution of 480^2 failed to outperform SINetV2 [23] with 352^2 in all metrics. This observation raises two critical questions: should high-resolution be used in concealed scenarios, and how can we develop an effective strategy for detecting concealed objects of varying sizes? We will propose potential solutions to these questions and present an interesting analysis of the COD10K in Section 5.5.

5.4 Quantitative analysis of NC4K

Compared to the CAMO dataset, the NC4K [24] dataset has a larger data scale and sample diversity, indicating subtle changes may have occurred. Table 5 presents quantitative results on the current largest COS testing dataset with 4121 samples. The benchmark includes 28 convolutional-based and four transformer-based approaches. Our observations are summarized as follows.

• CamoFormer-C [148] still outperformed all methods on NC4K. In contrast to the awkward situation observed on CAMO as described in Section 5.3, the ResNet-50

based CamoFormer-R [148] now performed better than two other competitors (i.e., SegMaR [136] and ZoomNet [134]) on NC4K. These results confirm the effectiveness of CamoFormer's decoder design in mapping latent features back to the prediction space, particularly for more complicated scenarios.

• DGNet [28] shows less promising results on the challenging NC4K dataset, possibly due to its restricted modeling capability with small model parameters. Nevertheless, this drawback provides an opportunity for modification since the model has a lightweight and simple architecture.

• While PopNet [149] may not perform well on small-scale CMAO datasets, it has demonstrated potential in the challenging NC4K dataset. This indicates that using an extra network to synthesize depth priors would be helpful for challenging samples. When compared to SINetV2 based on Res2Net-50 [23], PopNet has a heavier design (188.05M *vs.* 26.98M) and larger input resolution (512^2 *vs.* 352^2), but only improves the E_ϕ^{mn} value by 0.6%.

• Regarding the CamoFormer [148] model, there is now a noticeable difference in performance between its two variants. Specifically, the CamoFormer-S variant based on Swin-B lags behind while the CamoFormer-P variant based on PVTv2-B4 performs better.

5.5 Quantitative analysis of COD10K

In Table 6, we present a performance comparison of 36 competitors, including 32 convolutional-based models and four transformer-based models, on the COD10K dataset with diverse concealed samples. Based on our evaluation, we have made the following observations.

• CamoFormer-C [148], which has a robust backbone, remains the best-performing method among all

Table 6 Quantitative comparison on COD10K [20] testing set

Model	Pub/Year	Backbone	$\mathcal{S}_\alpha \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$E_\phi^{ad} \uparrow$	$E_\phi^{mn} \uparrow$	$E_\phi^{mx} \uparrow$	$F_\beta^{ad} \uparrow$	$F_\beta^{mn} \uparrow$	$F_\beta^{mx} \uparrow$
• *Convolution-based Backbone*											
SINet [20]	$CVPR_{20}$	ResNet-50	0.776	0.631	0.043	0.867	0.864	0.874	0.667	0.679	0.691
D2CNet [112]	TIE_{21}	ResNet-50	0.807	0.680	0.037	0.879	0.876	0.887	0.702	0.720	0.736
C2FNet [113]	$IJCAI_{21}$	Res2Net-50	0.813	0.686	0.036	0.886	0.890	0.900	0.703	0.723	0.743
TINet [115]	$AAAI_{21}$	ResNet-50	0.793	0.635	0.042	0.848	0.861	0.878	0.652	0.679	0.712
JSCOD [108]	$CVPR_{20}$	ResNet-50	0.809	0.684	0.035	0.882	0.884	0.891	0.705	0.721	0.738
LSR [24]	$CVPR_{21}$	ResNet-50	0.804	0.673	0.037	0.883	0.880	0.892	0.699	0.715	0.732
R-MGL [116]	$CVPR_{21}$	ResNet-50	0.814	0.666	0.035	0.865	0.852	0.890	0.681	0.711	0.738
S-MGL [116]	$CVPR_{21}$	ResNet-50	0.811	0.655	0.037	0.851	0.845	0.889	0.667	0.702	0.733
PFNet [25]	$CVPR_{21}$	ResNet-50	0.800	0.660	0.040	0.868	0.877	0.890	0.676	0.701	0.725
UGTR [117]	$ICCV_{21}$	ResNet-50	0.818	0.667	0.035	0.850	0.853	0.891	0.671	0.712	0.742
BAS [118]	$arXiv_{21}$	ResNet-34	0.802	0.677	0.038	0.869	0.855	0.870	0.707	0.715	0.729
NCHIT [119]	$CVIU_{22}$	ResNet-50	0.792	0.591	0.046	0.794	0.819	0.879	0.596	0.649	0.698
C2FNet-V2 [121]	$TCSVT_{22}$	Res2Net-50	0.811	0.691	0.036	0.890	0.887	0.896	0.718	0.725	0.742
CubeNet [122]	PR_{22}	ResNet-50	0.795	0.643	0.041	0.862	0.865	0.883	0.669	0.692	0.715
ERRNet [123]	PR_{22}	ResNet-50	0.786	0.630	0.043	0.845	0.867	0.886	0.646	0.675	0.702
TPRNet [124]	$TVCJ_{22}$	Res2Net-50	0.817	0.683	0.036	0.869	0.887	0.903	0.694	0.724	0.748
FAPNet [127]	TIP_{22}	Res2Net-50	0.822	0.694	0.036	0.875	0.888	0.902	0.707	0.731	0.758
BSANet [126]	$AAAI_{22}$	Res2Net-50	0.818	0.699	0.034	0.894	0.891	0.901	0.723	0.738	0.753
OCENet [130]	$WACV_{22}$	ResNet-50	0.827	0.707	0.033	0.885	0.894	0.905	0.718	0.741	0.764
BGNet [131]	$IJCAI_{22}$	Res2Net-50	0.831	0.722	0.033	0.902	0.901	0.911	0.739	0.753	0.774
PreyNet [132]	MM_{22}	ResNet-50	0.813	0.697	0.034	0.894	0.881	0.891	0.731	0.736	0.747
ZoomNet [134]	$CVPR_{22}$	ResNet-50	0.838	**0.729**	**0.029**	0.893	0.888	0.911	**0.741**	**0.766**	0.780
FDNet [135]	$CVPR_{22}$	Res2Net-50	**0.840**	**0.729**	0.030	**0.906**	**0.919**	**0.935**	0.728	0.757	**0.788**
SegMaR [136]	$CVPR_{22}$	ResNet-50	0.833	0.724	0.034	0.893	0.899	0.906	0.739	0.757	0.774
SINetV2 [23]	$TPAMI_{22}$	Res2Net-50	0.815	0.680	0.037	0.864	0.887	0.906	0.682	0.718	0.752
CamoFormer-C [148]	$arXiv_{23}$	ConvNeXt-B	**0.860**	**0.770**	**0.024**	**0.926**	**0.926**	**0.935**	**0.778**	**0.798**	**0.818**
CamoFormer-R [148]	$arXiv_{23}$	ResNet-50	0.838	0.724	**0.029**	0.900	**0.916**	**0.930**	0.721	0.753	0.786
PopNet [149]	$arXiv_{23}$	Res2Net-50	**0.851**	**0.757**	**0.028**	**0.910**	0.910	**0.919**	**0.771**	**0.786**	**0.802**
CRNet [142]	$AAAI_{23}$	ResNet-50	0.733	0.576	0.049	0.845	0.832	0.845	0.637	0.633	0.636
PFNet+ [26]	$Ssis_{23}$	ResNet-50	0.806	0.677	0.037	0.880	0.884	0.895	0.698	0.716	0.734
DGNet-S [28]	MIR_{23}	EfficientNet-B1	0.810	0.672	0.036	0.869	0.888	0.905	0.680	0.710	0.743
DGNet [28]	MIR_{23}	EfficientNet-B4	0.822	0.693	0.033	0.879	0.896	0.911	0.698	0.728	0.759
• *Transformer-based Backbone*											
DTINet [133]	$ICPR_{22}$	MiT-B5	0.824	0.695	**0.034**	0.881	0.896	0.911	0.702	0.726	0.754
CamoFormer-S[148]	$arXiv_{23}$	Swin-B	**0.862**	**0.772**	**0.024**	**0.932**	**0.931**	**0.941**	**0.780**	**0.799**	**0.818**
CamoFormer-P[148]	$arXiv_{23}$	PVTv2-B4	**0.869**	**0.786**	**0.023**	**0.931**	**0.932**	**0.939**	**0.794**	**0.811**	**0.829**
HitNet [143]	$AAAI_{23}$	PVTv2-B2	**0.871**	**0.806**	**0.023**	**0.936**	**0.935**	**0.938**	**0.818**	**0.823**	**0.838**

convolutional-based methods. Similar to its performance on NC4K, CamoFormer-R [148] has once again outperformed strong competitors with identical backbones such as SegMaR [136] and ZoomNet [134].

• Similar to its performance on the NC4K dataset, PopNet [149] achieved consistently competitive results on the COD10K dataset, ranking second only to CamoFormer-C [148]. We believe that prior knowledge of the depth of the scene plays a crucial role in enhancing the understanding of concealed environments. This insight will motivate us to investigate more intelligent ways to learn structural priors, such as incorporating multi-task learning or heuristic methods into our models.

• Notably, HitNet [143] achieved the highest performance on the COD10K benchmark, outperforming models with stronger backbones such as Swin-B and PVTv2-B4. To understand why this is the case, we calculated the average resolution of all samples in the CAMO (W = 693.89 and H = 564.22), NC4K (W = 709.19 and H = 529.61), and COD10K (W = 963.34 and H = 740.54) datasets. We found that the testing set for COD10K has the highest overall resolution, which suggests that models utilizing higher resolutions or multi-scale modeling would benefit from this characteristic. Therefore, HitNet is an excellent choice for detecting concealed objects in scenarios where high-resolution images are available.

5.6 Qualitative comparison

This section visually assesses the performance of current top models on challenging and complex samples that are prone to failure. We compare qualitative results predicted by ten groups of top-performing models, including six convolutional-based models (i.e., CamoFormer-C [148], DGNet [28], PopNet [149], ZoomNet [134], FDNet [135] and SINetV2 [23]), two transformer-based models (i.e., CamoFormer-S [148] and HitNet [143]), as well as two other competitors (i.e., the earliest baseline SINet [20] and a weakly-supervised model CRNet [142]). All samples are selected from the COD10K testing dataset according to seven fine-grained attributes. The qualitative comparison is presented in Fig. 4, revealing several interesting findings.

• The attribute of multiple objects (MO) poses a challenge due to the high false-negative rate in current top-performing models. As depicted in the first column of Fig. 4, only two out of ten models could locate the white flying bird approximately, as indicated by the red circle in the GT mask. These two models are CamoFormer-S [148], which employs a robust transformer-based encoder, and FDNet [135], which utilizes a frequency-domain learning strategy.

• The models we tested can accurately detect big objects (BO) by precisely locating the target's main part. However, these models struggle to identify smaller details such as the red circles highlighting the toad's claws in the second column of Fig. 4.

• The small object (SO) attribute presents a challenge as it only occupies a small area in the image, typically less than 10% of the total pixels as reported by COD10K [20]. As shown in the third column of Fig. 4, only two models (CamoFormer-S and CamoFormer-C [148]) can detect a cute cat lying on the ground at a distance. Such a difficulty arises for two main reasons. First, models struggle to differentiate small objects from complex backgrounds or other irrelevant objects in an image. Second, detectors may miss small regions due to down-sampling operations caused by low-resolution inputs.

• The out-of-view (OV) attribute refers to objects partially outside the image boundaries, leading to incomplete representation. To address this issue, a model should have a better holistic understanding of the concealed scene. As shown in the fourth column of Fig. 4, both CamoFormer-C [148] and FDNet [135] can handle the OV attribute and maintain the object's integrity. However, two transformer-based models failed to do so. This observation has inspired us to explore more efficient methods, such as local modeling within convolutional frameworks and cross-domain learning strategies.

• The shape complexity (SC) attribute indicates that an object contains thin parts, such as an animal's foot. In the fifth column of Fig. 4, the stick insect's feet are a good example of this complexity, being elongated and slender and thus difficult to predict accurately. Only HitNet [143] with high-resolution inputs can predict a right-bottom foot (indicated by a red circle).

• The attribute of occlusion (OC) refers to the partial occlusion of objects, which is a common challenge in general scenes [184]. In Fig. 4, for example, the sixth column shows two owls partially occluded by a wire fence, causing their visual regions to be separated. Unfortunately, most of the models presented were unable to handle such cases.

• The indefinable boundary (IB) attribute is difficult to address due to its uncertainty between the foreground and background. As shown in the last column of Fig. 4, a matting-level sample.

• In the last two rows of Fig. 4, we display the predictions generated by SINet [20], which was our earliest baseline model. Current models have significantly improved location accuracy, boundary details, and other aspects. Additionally, CRNet [142], a weakly-supervised method with only weak label supervision, can effectively locate target objects to meet satisfactory standards.

6 Discussion and outlook

Based on our literature review and experimental analysis, we discuss five challenges and potential CSU-related directions in this section.

• *Annotation-efficient learning.* Deep learning techniques have significantly advanced the field of CSU. However, conventional supervised deep learning is data-hungry

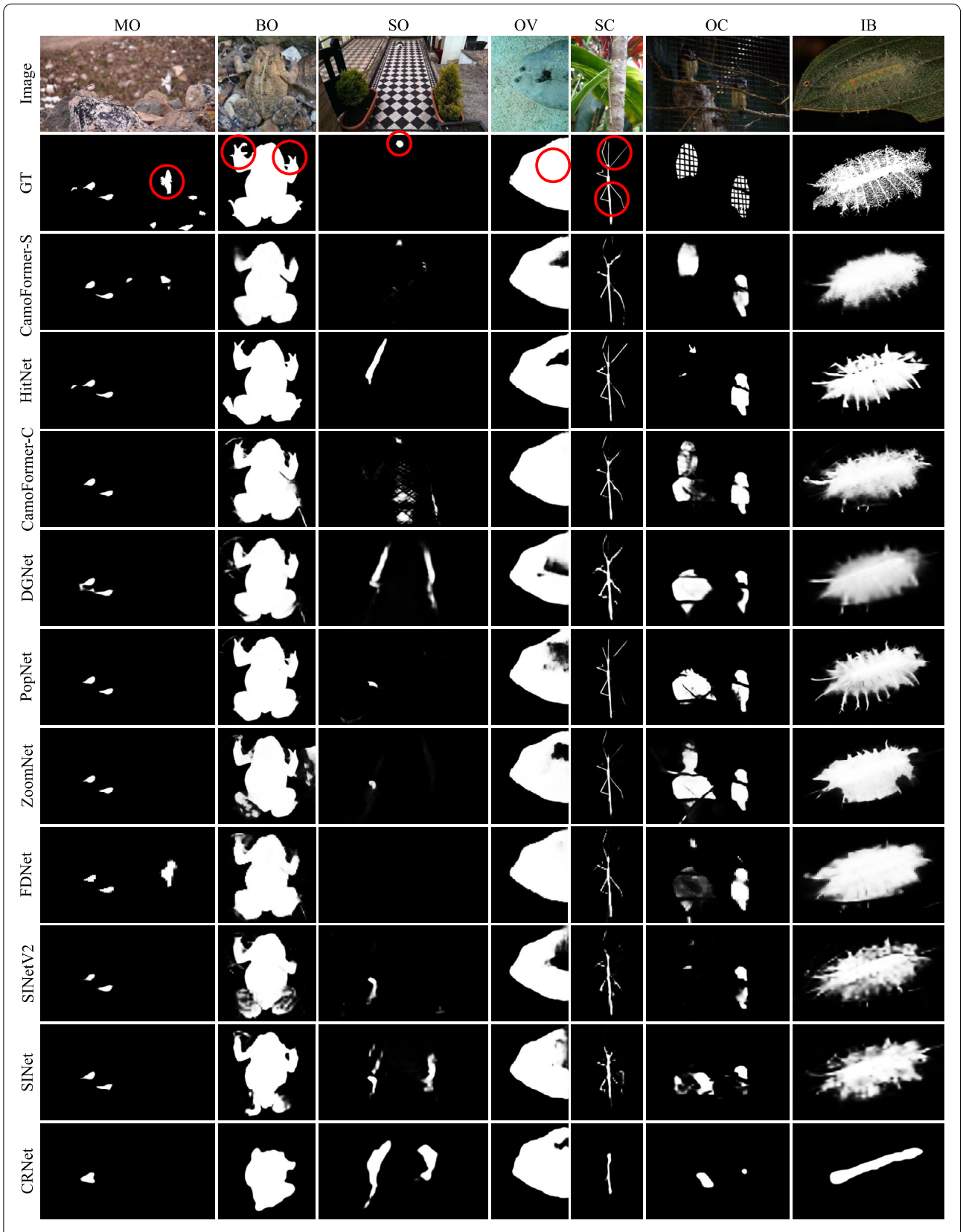

Figure 4 Qualitative results of ten COS approaches. More descriptions on visual attributes in each column refer to Section 5.6

and resource-consuming. In practical scenarios, we hope the models can work on limited resources and have good generalizability. Thus developing effective learning strategies for CSU tasks is a promising direction, e.g., the weakly-supervised strategy in CRNet [142].

• *Domain adaptation.* Camouflaged samples are generally collected from natural scenes. Thus, deploying the models to detect concealed objects in auto-driving scenarios is challenging. Recent practice demonstrates that various techniques can be used to alleviate this problem, e.g., domain adaptation [185, 186], transfer learning [187], few-shot learning [188], and meta-learning [189].

• *High-fidelity synthetic dataset.* To alleviate algorithmic biases, increasing the diversity and scale of data is crucial. The rapid development of artificial intelligence generated content (AIGC) [190] and deep generative models, such as generative adversarial networks [191–193] and diffusion models [194, 195], is making it easier to create synthetic data for general domains. Recently, to address the scarcity of multi-pattern training images, Luo et al. [107] proposed a diffusion-based image generation framework that generates salient objects on a camouflaged sample while preserving its original label. Therefore, a model should be capable of distinguishing between camouflaged and salient objects to achieve a robust feature representation.

• *Neural architecture search.* Automatic network architecture search (NAS) is a promising research direction that can discover optimal network architectures for superior performance on a given task. In the context of concealment, NAS can identify more effective network architectures to handle complex background scenes, highly variable object appearances, and limited labeled data. This can lead to the development of more efficient and effective network architectures, resulting in improved accuracy and efficiency. Combining NAS with other research directions, such as domain adaptation and data-efficient learning, can further enhance the understanding of concealed scenes. These avenues of exploration hold significant potential for advancing the state-of-the-art and warrant further investigation in future research.

• *Large model and prompt engineering.* This topic has gained popularity and has even become a direction for the natural language processing community. Recently, the segment anything model (SAM) [196] has revolutionized computer vision algorithms, although it has limitations [197] in unprompted settings on several concealed scenarios. One can leverage the prompt engineering paradigm to simplify workflows using a well-trained robust encoder and task-specific adaptions, such as task-specific prompts and multi-task prediction heads. This approach is expected to become a future trend within the computer vision community. Large language models (LLMs) have brought both new opportunities and challenges to AI, moving towards artificial general intelligence further. However, it is challenging for academia to train the resource-consuming large models. There could be a promising paradigm in which the state-of-the-art deep learning CSU models are used as the domain experts, and the large models could work as an external component to assist the expert models by providing an auxiliary decision, representation, etc.

7 Defect segmentation dataset

Industrial defects usually originate from the undesirable production process, e.g., mechanical impact, workpiece friction, chemical corrosion, and other unavoidable physical conditions, whose external visual form is usually with unexpected patterns or outliers, e.g., surface scratches, spots, holes on industrial devices; color difference, indentation on fabric surface; impurities, breakage, stains on the material surface, etc. Although previous works have achieved promising advances in identifying visual defects by vision-based techniques, such as classification [198–200], detection [201–203], and segmentation [204–206], these techniques work on the assumption that defects are easily detected, but they ignore those challenging defects that are "seamlessly" embedded in their materials' surroundings. With this, we elaborately collected a new multi-scene benchmark, named CDS2K, for the concealed defect segmentation task, whose samples were selected from existing industrial defect databases.

7.1 Dataset organization

To create a dataset of superior quality, we established three principles for selecting data: (1) The chosen sample should include at least one defective region, which will serve as a positive example. (2) The defective regions should have a pattern similar to the background, making them difficult to identify. (3) We also select normal cases as negative examples to provide a contrasting perspective with the positive examples. These samples were selected from the following well-known defect segmentation databases.

• MVTecAD[4] [207, 208] contains several positive and negative samples for unsupervised anomaly detection. We manually selected 748 positive and 746 negative samples with concealed patterns from two main categories: (1) object category as in the 1st row of Fig. 5: pill, screw, tile, transistor, wood, and zipper. (2) texture category as in the 2nd row of Fig. 5: bottle, capsule, carpet, grid, leather, and metal nut. The number of positive/negative samples is shown with yellow circles in Fig. 5

• NEU[5] provides three different databases: oil pollution defect images [209] (OPDI), spot defect images [210] (SDI), and steel pit defect images [211] (SPDI). As displayed in the third row (green circles) of Fig. 5, we selected

[4] https://www.mvtec.com/company/research/datasets/mvtec-ad.

[5] http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm.

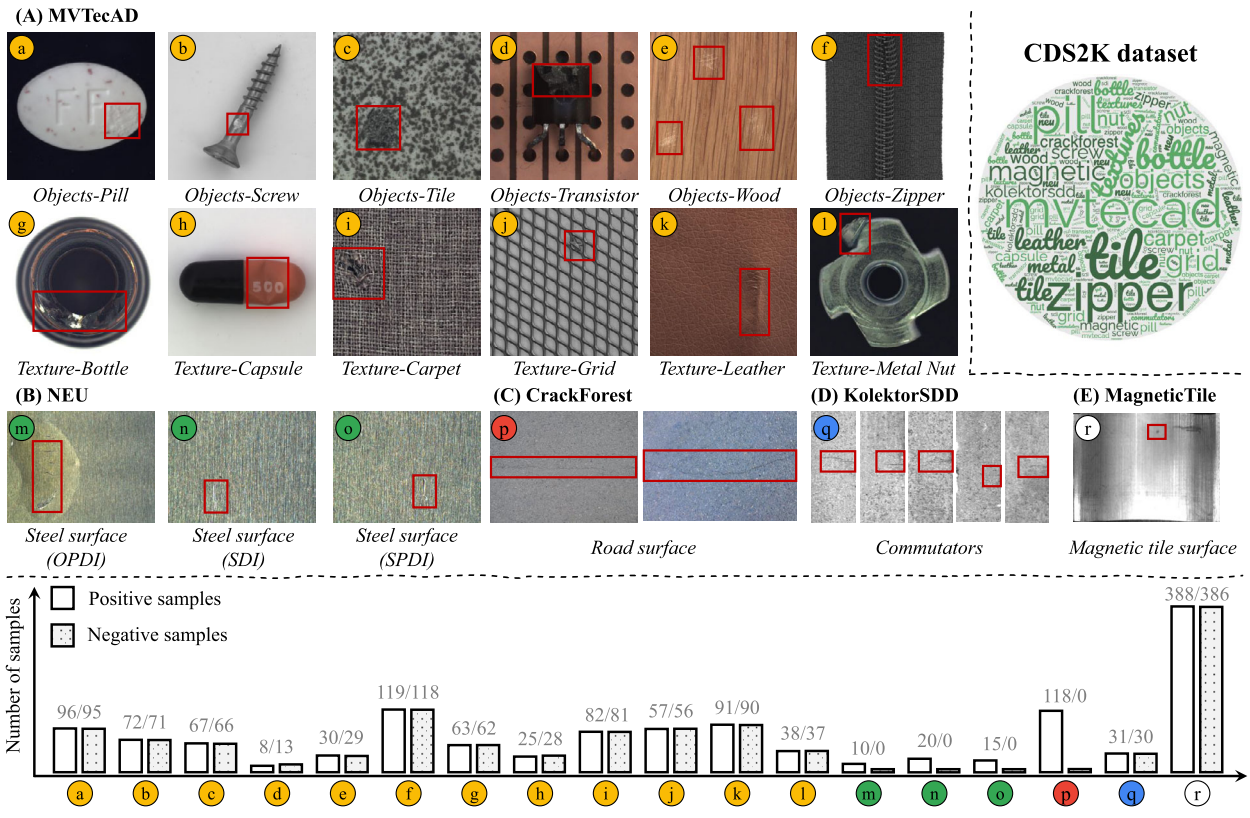

Figure 5 Sample gallery of our CDS2K. It is collected from five sub-databases: (a)–(l) MVTecAD, (m)–(o) NEU, (p) CrackForest, (q) KolektorSDD, and (r) MagneticTile. The defective regions are highlighted with red rectangles. (Top-Right) Word cloud visualization of CDS2K. (Bottom) The statistic number of positive/negative samples of each category in our CDS2K

10, 20, and 15 positive samples from these databases separately.

• CrackForest[6] [212, 213] is a densely-annotated road crack image database for the health monitoring of urban road surface. We selected 118 samples with concealed patterns from them, and the samples are shown in the third row (red circle) of Fig. 5.

• KolektorSDD[7] [205] collected and annotated by Kolektor Group, which contains several defective and non-defective surfaces from the controlled industrial environment in a real-world case. We manually selected 31 positive and 30 negative samples with concealed patterns, and the samples are shown in the third row (blue circle) of Fig. 5.

• MagneticTile defect[8] [214] datasets contain six common magnetic tile defects and corresponding dense annotations. We picked 388 positive and 386 negative examples, displayed as white circles in Fig. 5.

[6]https://github.com/cuilimeng/CrackForest-dataset.

[7]https://www.vicos.si/resources/kolektorsdd/.

[8]https://github.com/abin24/Magnetic-tile-defect-datasets.

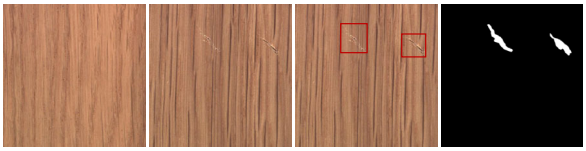

Figure 6 Visualization of different annotations. We selected a group of images from the MVTecAD, including a negative (a) and a positive (b) sample. Corresponding annotations are provided: category (scratches on wood) and defect locations: bounding box (c) and segmentation mask (d)

7.2 Dataset description

The CDS2K comprises 2492 samples, consisting of 1330 positive and 1162 negative instances. Three different human-annotated labels are provided for each sample – category, bounding box, and pixel-wise segmentation mask. Figure 6 illustrates examples of these annotations. The average ratio of defective regions for each category is presented in Table 7, which indicates that most of the defective regions are relatively small.

Table 7 Statistics of positive samples in CDS2K. The region ratio is calculated by r = defective pixels/all pixels for a given image. Of note, we only count the number of positive samples in five sub-datasets

Category		$0\% < r < 1\%$	$1\% \leq r < 10\%$	$10\% \leq r < 20\%$	$20\% \leq r < 30\%$	$30\% \leq r < 40\%$	$40\% \leq r < 50\%$	*Total*
MVTecAD	Objects-Pill	41	55	0	0	0	0	**96**
	Objects-Screw	71	1	0	0	0	0	**72**
	Objects-Tile	0	30	28	7	2	0	**67**
	Objects-Transistor	1	7	0	0	0	0	**8**
	Objects-Wood	2	26	2	0	0	0	**30**
	Objects-Zipper	16	102	1	0	0	0	**119**
	Texture-Bottle	3	39	20	1	0	0	**63**
	Texture-Capsule	17	8	0	0	0	0	**25**
	Texture-Carpet	37	45	0	0	0	0	**82**
	Texture-Grid	39	18	0	0	0	0	**57**
	Texture-Leather	70	21	0	0	0	0	**91**
	Texture-Metal Nut	6	31	1	0	0	0	**38**
NEU	OPDI	10	0	0	0	0	0	**10**
	SDI	20	0	0	0	0	0	**20**
	SPDI	15	0	0	0	0	0	**15**
CrackForest		28	90	0	0	0	0	**118**
KolektorSDD		31	0	0	0	0	0	**31**
MagneticTile defect		216	70	27	27	24	24	**388**
Total		**623**	**543**	**79**	**35**	**26**	**24**	**1330**

Table 8 Quantitative comparison of the positive samples of CDS2K

Model	Pub/Year	Backbone	$\mathcal{S}_\alpha \uparrow$	$F_\beta^w \uparrow$	$\mathcal{M} \downarrow$	$E_\phi^{ad} \uparrow$	$E_\phi^{mn} \uparrow$	$E_\phi^{mx} \uparrow$	$F_\beta^{ad} \uparrow$	$F_\beta^{mn} \uparrow$	$F_\beta^{mx} \uparrow$
SINetV2 [23]	$TPAMI_{22}$	Res2Net-50	0.551	0.215	0.102	0.509	0.567	0.597	0.223	0.248	0.258
HitNet [143]	$AAAI_{23}$	PVTv2-B2	0.563	0.276	0.118	0.574	0.564	0.570	0.298	0.298	0.299
DGNet [28]	MIR_{23}	EfficientNet-B4	0.578	0.258	0.089	0.552	0.569	0.579	0.274	0.291	0.297
CamoFormer-P [148]	$arXiv_{23}$	PVTv2-B4	0.589	0.298	0.100	0.590	0.588	0.596	0.330	0.329	0.339

7.3 Evaluation on CDS2K

Here, we evaluate the generalizability of current cutting-edge COS models on the positive samples of CDS2K. Regrading the code availability, we chose four top-performing COS approaches: SINetV2 [23], DGNet [28], CamoFormer-P [148], and HitNet [143]. As reported in Table 8, our observations indicate that these models are not effective in handling cross-domain samples, highlighting the need for further exploration of the domain gap between natural scenes and downstream applications.

8 Conclusion

This paper aims to provide an overview of deep learning techniques tailored for concealed scene understanding (CSU). To help readers view the global landscape of this field, we have made four contributions. First, we provide a detailed survey of CSU, which includes its background, taxonomy, task-specific challenges, and advances in the deep learning era. To the best of our knowledge, this survey is the most comprehensive one to date. Second, we have created the largest and most up-to-date benchmark for concealed object segmentation (COS), which is a foundational and prosperous direction at CSU. This benchmark allows for a quantitative comparison of state-of-the-art techniques. Third, we collected the largest concealed defect segmentation dataset, CDS2K, by including hard cases from diverse industrial scenarios. We have also constructed a comprehensive benchmark to evaluate the generalizability of deep CSU in practical scenarios. Finally, we discuss open problems and potential directions for this community. We aim to encourage further research and development in this area.

We would conclude from the following perspectives. (1) *Model.* The most common practice is based on the architecture of sharing UNet, which is enhanced by various attention modules. In addition, injecting extra priors and/or introducing auxiliary tasks improve the performance, while there are many potential problems to explore. (2) *Training.* Fully-supervised learning is the mainstream strategy in COS, but few researchers have addressed the challenge caused by insufficient data or labels. CRNet [142] is a good attempt to alleviate this issue. (3) *Dataset.* The existing datasets are still not sufficiently large and diverse. This community needs more concealed samples involving more domains (e.g., autonomous driving and clinical diagnosis). (4) *Performance.* Transformer and ConvNext based models outperform other competitors by a clear margin. Cost-performance tradeoff is still under-studied, for which DGNet [28] is a good attempt.

(5) *Metric.* There are no well-defined metrics that can consider different camouflage degrees of different data to provide a comprehensive evaluation. This causes unfair comparisons.

Additionally, existing CSU methods focus on the appearance attributes of concealed scenes (e.g., color, texture, and boundary) to distinguish concealed objects without sufficient perception and output from the semantic perspective (e.g., relationships between objects). However, semantics is a good tool for bridging the human and machine intelligence gap. Therefore, beyond the visual space, semantic level awareness is key to the next-generation concealed visual perception. In the future, CSU models should incorporate various semantic associations, including integrating high-level semantics, learning vision-language knowledge [215], and modeling interactions across objects.

We hope that this survey provides a detailed overview for new researchers, presents a convenient reference for relevant experts, and encourages future research.

Acknowledgements
We want to thank Yu-Cheng Chou for investigating relevant literature. The authors express their gratitude to the anonymous reviewers and the editor, whose valuable feedback greatly improved the quality of this manuscript.

Funding
Deng-Ping Fan and Christos Sakaridis are funded by Toyota Motor Europe (research project TRACE-Zürich).

Abbreviations
CSU, concealed scene understanding; COS, concealed object segmentation; COL, concealed object localization; CIR, concealed instance ranking; CIS, concealed instance segmentation; COC, concealed object counting; VCOD, video concealed object detection; VCOS, video concealed object segmentation; SOD, salient object detection; VSOD, video salient object detection; IOC, indiscernible object counting; AIGC, artificial intelligence generated content; NAS, network architecture search; SAM, segment anything model; LLMs, large language models.

Availability of data and materials
Our sources including code and datasets can be accessed via GitHub: https://github.com/DengPingFan/CSU.

Code availability
Our code and datasets are available at https://github.com/DengPingFan/CSU, which will be updated continuously to watch and summarize the advancements in this rapidly evolving field.

Declarations

Competing interests
All authors certify that they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

Author contributions
Project Lead: D-PF; Conceptualization: D-PF, M-MC; Formal analysis and investigation: G-PJ, D-PF, PX; Writing (original draft preparation): G-PJ, D-PF; Writing (literature search and data analysis): D-PF, G-PJ, PX; Writing (critical revision): D-PF, PX, M-MC, CS; Supervision: M-MC, LVG. All authors read and approved the final manuscript.

Author details
[1]CVL, ETH Zurich, Zurich 8092, Switzerland. [2]CECC, ANU, Canberra 0200, Australia. [3]EE, Tsinghua University, Beijing 100084, China. [4]CS, Nankai University, Tianjin 300350, China.

Received: 21 April 2023 Revised: 10 July 2023 Accepted: 10 July 2023
Published online: 14 August 2023

References

1. Fan, D.-P., Zhang, J., Xu, G., Cheng, M.-M., & Shao, L. (2023). Salient objects in clutter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(2), 2344–2366.
2. Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6230–6239). Los Alamitos: IEEE.
3. Ji, G.-P., Xiao, G., Chou, Y.-C., Fan, D.-P., Zhao, K., Chen, G., et al. (2022). Video polyp segmentation: a deep learning perspective. *Management International Review*, *19*(6), 531–549.
4. Ji, G.-P., Zhang, J., Campbell, D., Xiong, H., & Barnes, N. (2023). *Rethinking polyp segmentation from an out-of-distribution perspective*. arXiv preprint arXiv:2306.07792.
5. Fan, D.-P., Zhou, T., Ji, G.-P., Zhou, Y., Chen, G., Fu, H., et al. (2020). Inf-Net: automatic COVID-19 lung infection segmentation from CT images. *IEEE Transactions on Medical Imaging*, *39*(8), 2626–2637.
6. Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., et al. (2019). PestNet: an end-to-end deep learning approach for large-scale multi-class pest detection and classification. *IEEE Access*, *7*, 45301–45312.
7. Rizzo, M., Marcuzzo, M., Zangari, A., Gasparetto, A., & Albarelli, A. (2023). Fruit ripeness classification: a survey. *Artificial Intelligence in Agriculture*, *7*, 44–57.
8. Chu, H.-K., Hsu, W.-H., Mitra, N. J., Cohen-Or, D., Wong, T.-T., & Lee, T.-Y. (2010). Camouflage images. *ACM Transactions on Graphics*, *29*(4), 51.
9. Boult, T. E., Micheals, R. J., Gao, X., & Eckmann, M. (2001). Into the woods: visual surveillance of noncooperative and camouflaged targets in complex outdoor settings. *Proceedings of the IEEE*, *89*(10), 1382–1402.
10. Conte, D., Foggia, P., Percannella, G., Tufano, F., & Vento, M. (2009). An algorithm for detection of partially camouflaged people. In S. Tubaro & J.-L. Dugelay (Eds.), *Proceedings of the sixth IEEE international conference on advanced video and signal based surveillance* (pp. 340–345). Los Alamitos: IEEE.
11. Yin, J., Han, Y., Hou, W., & Li, J. (2011). Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*, *15*, 2201–2205.
12. Kim, S. (2015). Unsupervised spectral-spatial feature selection-based camouflaged object detection using VNIR hyperspectral camera. *The Scientific World Journal*, *2015*, 1–8.
13. Zhang, X., Zhu, C., Wang, S., Liu, Y., & Ye, M. (2016). A Bayesian approach to camouflaged moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(9), 2001–2013.
14. Galun, M., Sharon, E., Basri, R., & Brandt, A. (2003). Texture segmentation by multiscale aggregation of filter responses and shape elements. In *2003 IEEE international conference on computer vision* (pp. 716–723). Los Alamitos: IEEE.
15. Tankus, A., & Yeshurun, Y. (1998). Detection of regions of interest and camouflage breaking by direct convexity estimation. In *1998 IEEE workshop on visual surveillance* (pp. 1–7). Los Alamitos: IEEE.
16. Tankus, A., & Yeshurun, Y. (2001). Convexity-based visual camouflage breaking. *Computer Vision and Image Understanding*, *82*(3), 208–237.
17. Mittal, A., & Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 302–309). Los Alamitos: IEEE.
18. Liu, Z., Huang, K., & Tan, T. (2012). Foreground object detection using top-down information based on EM framework. *IEEE Transactions on Image Processing*, *21*(9), 4204–4217.
19. Li, S., Florencio, D., Zhao, Y., Cook, C., & Li, W. (2017). Foreground detection in camouflaged scenes. In *2017 IEEE international conference on image processing* (pp. 4247–4251). Los Alamitos: IEEE.
20. Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J., & Shao, L. (2020). Camouflaged object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2774–2784). Los Alamitos: IEEE.

21. Le, T.-N., Nguyen, T. V., Nie, Z., Tran, M.-T., & Sugimoto, A. (2019). Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, *184*, 45–56.
22. Zhang, Q., Yin, G., Nie, Y., & Zheng, W.-S. (2020). Deep camouflage images. In *Proceedings of the 34th AAAI conference on artificial intelligence* (pp. 12845–12852). Menlo Park: AAAI Press.
23. Fan, D.-P., Ji, G.-P., Cheng, M.-M., & Shao, L. (2022). Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(10), 6024–6042.
24. Lv, Y., Zhang, J., Dai, Y., Li, A., Liu, B., Barnes, N., et al. (2021). Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11591–11601). Los Alamitos: IEEE.
25. Mei, H., Ji, G.-P., Wei, Z., Yang, X., Wei, X., & Fan, D.-P. (2021). Camouflaged object segmentation with distraction mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8772–8781). Los Alamitos: IEEE.
26. Mei, H., Yang, X., Zhou, Y., Ji, G.-P., Wei, X., & Fan, D.-P. (2023). Distraction-aware camouflaged object segmentation. *SCIENTIA SINICA Informationis*. Advance online publication. https://doi.org/10.1360/SSI-2022-0138
27. Yu, L., Mei, H., Dong, W., Wei, Z., Zhu, L., Wang, Y., et al. (2022). Progressive glass segmentation. *IEEE Transactions on Image Processing*, *31*, 2920–2933.
28. Ji, G.-P., Fan, D.-P., Chou, Y.-C., Dai, D., Liniger, A., & Van Gool, L. (2023). Deep gradient learning for efficient camouflaged object detection. *Management International Review*, *20*(1), 92–108.
29. Kulchandani, J. S., & Dangarwala, K. J. (2015). Moving object detection: review of recent research trends. In *2015 international conference on pervasive computing* (pp. 1–5). Los Alamitos: IEEE.
30. Mondal, A. (2020). Camouflaged object detection and tracking: a survey. *International Journal of Image and Graphics*, *20*(4), 2050028.
31. Bi, H., Zhang, C., Wang, K., Tong, J., & Zheng, F. (2022). Rethinking camouflaged object detection: models and datasets. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(9), 5708–5724.
32. Caijuan, S., Bijuan, R., Ziwen, W., Jinwei, Y., & Ze, S. (2022). Survey of camouflaged object detection based on deep learning. *Journal of Frontiers of Computer Science and Technology*, *16*(12), 2734.
33. Lv, Y., Zhang, J., Dai, Y., Li, A., Barnes, N., & Fan, D.-P. (2023). Towards deeper understanding of camouflaged object detection. *IEEE transactions on circuits and systems for video technology*, *33*(7), 3462–3476.
34. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *2017 IEEE international conference on computer vision* (pp. 2980–2988). Los Alamitos: IEEE.
35. Pei, J., Cheng, T., Fan, D.-P., Tang, H., Chen, C., & Van Gool, L. (2022). Osformer: one-stage camouflaged instance segmentation with transformers. In S. Avidan, G. J. Brostow, M. Cissé, et al. (Eds.), *Proceedings of the 17th European conference of computer vision* (pp. 19–37). Berlin: Springer.
36. Le, T.-N., Cao, Y., Nguyen, T.-C., Le, M.-Q., Nguyen, K.-D., Do, T.-T., et al. (2022). Camouflaged instance segmentation in-the-wild: dataset, method, and benchmark suite. *IEEE Transactions on Image Processing*, *31*, 287–300.
37. Xie, E., Wang, W., Ding, M., Zhang, R., & Luo, P. (2021). Polarmask++: enhanced polar representation for single-shot instance segmentation and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5385–5400.
38. Chen, H., Sun, K., Tian, Z., Shen, C., Huang, Y., & Blendmask, Y. Y. (2020). Top-down meets bottom-up for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8570–8578). Los Alamitos: IEEE.
39. Sun, G., An, Z., Liu, Y., Liu, C., Sakaridis, C., Fan, D.-P., et al. (2023). Indiscernible object counting in underwater scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13791–13801). Los Alamitos: IEEE.
40. Lamdouar, H., Yang, C., Xie, W., & Zisserman, A. (2020). Betrayed by motion: camouflaged object discovery via motion segmentation. In H. Ishikawa, C.-L. Liu, T. Pajdla, et al. (Eds.), *Proceedings of the 15th Asian conference on computer vision* (pp. 488–503). Berlin: Springer.
41. Jiao, L., Zhang, R., Liu, F., Yang, S., Hou, B., Li, L., et al. (2022). New generation deep learning for video object detection: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(8), 3195–3215.
42. Cheng, X., Xiong, H., Fan, D.-P., Zhong, Y., Harandi, M., Drummond, T., et al. (2022). Implicit motion handling for video camouflaged object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13854–13863). Los Alamitos: IEEE.
43. Fan, D.-P., Cheng, M.-M., Liu, J.-J., Gao, S.-H., Hou, Q., & Borji, A. (2018). Salient objects in clutter: bringing salient object detection to the foreground. In V. Ferrari, M. Hebert, C. Sminchisescu, et al. (Eds.), *Proceeding of the 15th European conference on computer vision* (pp. 196–212). Berlin: Springer.
44. He, S., Lau, R. W. H., Liu, W., Huang, Z., & Yang, Q. (2015). SuperCNN: a superpixelwise convolutional neural network for salient object detection. *International Journal of Computer Vision*, *115*(3), 330–344.
45. Li, G., & Yu, Y. (2015). Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5455–5463). Los Alamitos: IEEE.
46. Wang, L., Lu, H., Ruan, X., & Yang, M.-H. (2015). Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3183–3192). Los Alamitos: IEEE.
47. Kim, J., & Pavlovic, V. (2016). A shape-based approach for salient object detection using deep learning. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer Vision* (pp. 455–470). Berlin: Springer.
48. Zeng, Y., Zhang, P., Zhang, J., Lin, Z., & Lu, H. (2019). Towards high-resolution salient object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 7233–7242). Los Alamitos: IEEE.
49. Liu, N., & Dhsnet, J. H. (2016). Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 678–686). Los Alamitos: IEEE.
50. Wu, Z., Su, L., & Huang, Q. (2019). Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3907–3916). Los Alamitos: IEEE.
51. Zhang, P., Wang, D., Lu, H., Wang, H., & Yin, B. (2017). Learning uncertain convolutional features for accurate saliency detection. In *2017 IEEE international conference on computer vision* (pp. 212–221). Los Alamitos: IEEE.
52. Hou, Q., Cheng, M.-M., Hu, X., Borji, A., Tu, Z., & Torr, P. H. S. (2019). Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(4), 815–828.
53. Zhuge, M., Fan, D.-P., Liu, N., Zhang, D., Xu, D., & Shao, L. (2023). Salient object detection via integrity learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 3738–3752.
54. Liu, Y., Zhang, Q., Zhang, D., & Han, J. (2019). Employing deep part-object relationships for salient object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 1232–1241). Los Alamitos: IEEE.
55. Qi, Q., Zhao, S., Shen, J., & Lam, K.-M. (2019). Multi-scale capsule attention-based salient object detection with multi-crossed layer connections. In *IEEE international conference on multimedia and expo (1762-1767)*. Los Alamitos: IEEE.
56. Liu, N., Zhang, N., Wan, K., Shao, L., & Han, J. (2021). Visual saliency transformer. In *2021 IEEE/CVF international conference on computer vision* (pp. 4702–4712). Los Alamitos: IEEE.
57. Li, G., & Yu, Y. (2016). Deep contrast learning for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 478–487). Los Alamitos: IEEE.
58. Tang, Y., & Wu, X. (2016). Saliency detection via combining region-level and pixel-level predictions with CNNs. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer Vision* (pp. 809–825). Berlin: Springer.
59. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., et al. (2017). Learning to detect salient objects with image-level supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3796–3805). Los Alamitos: IEEE.
60. Li, G., Xie, Y., & Lin, L. (2018). Weakly supervised salient object detection using image labels. In S. A. McIlraith, & K. Q. Weinberger (Eds.), *Proceedings of the 32nd AAAI conference on artificial intelligence* (pp. 7024–7031). Menlo Park: AAAI Press.
61. Cao, C., Huang, Y., Wang, Z., Wang, L., Xu, N., & Tan, T. (2018). Lateral inhibition-inspired convolutional neural network for visual attention and saliency detection. In S. A. McIlraith, & K. Q. Weinberger (Eds.), *Proceedings of the 32nd AAAI conference on artificial intelligence* (pp. 6690–6697). Menlo Park: AAAI Press.

62. Li, B., Sun, Z., & Supervae, Y. G. (2019). Superpixelwise variational autoencoder for salient object detection. In *Proceedings of the 33rd AAAI conference on artificial intelligence* (pp. 8569–8576). Menlo Park: AAAI Press.
63. Zeng, Y., Zhuge, Y., Lu, H., Zhang, L., Qian, M., & Yu, Y. (2019). Multi-source weak supervision for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6074–6083). Los Alamitos: IEEE.
64. Zhang, D., Han, J., & Zhang, Y. (2017). Supervision by fusion: towards unsupervised learning of deep salient object detector. In *2017 IEEE international conference on computer vision* (pp. 4068–4076). Los Alamitos: IEEE.
65. Zhang, J., Zhang, T., Dai, Y., Harandi, M., & Hartley, R. (2018). Deep unsupervised saliency detection: a multiple noisy labeling perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9029–9038). Los Alamitos: IEEE.
66. Shin, G., Albanie, S., & Xie, W. (2022). Unsupervised salient object detection with spectral cluster voting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3970–3979). Los Alamitos: IEEE.
67. He, S., Jiao, J., Zhang, X., Han, G., & Lau, R. W. (2017). Delving into salient object subitizing and detection. In *2017 IEEE international conference on computer vision* (pp. 1059–1067). Los Alamitos: IEEE.
68. Islam, M. A., Kalash, M., & Bruce, N. D. B. (2018). Revisiting salient object detection: simultaneous detection, ranking, and subitizing of multiple salient objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7142–7150). Los Alamitos: IEEE.
69. Wang, W., Shen, J., Dong, X., & Borji, A. (2018). Salient object detection driven by fixation prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1711–1720). Los Alamitos: IEEE.
70. Kruthiventi, S. S. S., Gudisa, V., Dholakiya, J. H., & Babu, R. V. (2016). Saliency unified: a deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5781–5790). Los Alamitos: IEEE.
71. Zeng, Y., Zhuge, Y., Lu, H., & Zhang, L. (2019). Joint learning of saliency detection and weakly supervised semantic segmentation. In *2019 IEEE/CVF international conference on computer vision* (pp. 7222–7232). Los Alamitos: IEEE.
72. Wang, L., Wang, L., Lu, H., Zhang, P., & Ruan, X. (2016). Saliency detection with recurrent fully convolutional networks. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer Vision* (pp. 825–841). Berlin: Springer.
73. Li, X., Yang, F., Cheng, H., Liu, W., & Shen, D. (2018). Contour knowledge transfer for salient object detection. In V. Ferrari, M. Hebert, C. Sminchisescu, et al. (Eds.), *Proceedings of the 15th European conference on computer Vision* (pp. 370–385). Berlin: Springer.
74. Wang, W., Zhao, S., Shen, J., Hoi, S. C., & Borji, A. (2019). Salient object detection with pyramid attention and salient edges. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1448–1457). Los Alamitos: IEEE.
75. Liu, J.-J., Hou, Q., Cheng, M.-M., Feng, J., & Jiang, J. (2019). A simple pooling-based design for real-time salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3917–3926). Los Alamitos: IEEE.
76. Zhao, J.-X., Liu, J.-J., Fan, D.-P., Cao, Y., Yang, J., & Cheng, M.-M. (2019). EGNet: edge guidance network for salient object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 8778–8787). Los Alamitos: IEEE.
77. Su, J., Li, J., Zhang, Y., Xia, C., & Tian, Y. (2019). Selectivity or invariance: boundary-aware salient object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 3798–3807). Los Alamitos: IEEE.
78. Zhang, L., Zhang, J., Lin, Z., Lu, H., & He, Y. (2019). CapSal: leveraging captioning to boost semantics for salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6024–6033). Los Alamitos: IEEE.
79. Li, G., Xie, Y., Lin, L., & Yu, Y. (2017). Instance-level salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 247–256). Los Alamitos: IEEE.
80. Tian, X., Xu, K., Yang, X., Yin, B., & Lau, R. W. (2022). Learning to detect instance-level salient objects using complementary image labels. *International Journal of Computer Vision*, *130*(3), 729–746.
81. Fan, R., Cheng, M.-M., Hou, Q., Mu, T.-J., Wang, J., & Hu, S.-M. (2019). S4Net: single stage salient-instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6103–6112). Los Alamitos: IEEE.
82. Wu, Y.-H., Liu, Y., Zhang, l., Gao, W., & Cheng, M.-M. (2021). Regularized densely-connected pyramid network for salient instance segmentation. *IEEE Transactions on Image Processing*, *30*, 3897–3907.
83. Borji, A., & Itti, L. (2012). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 185–207.
84. Borji, A. (2019). Saliency prediction in the deep learning era: successes and limitations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(2), 679–700.
85. Fan, D.-P., Li, T., Lin, Z., Ji, G.-P., Zhang, D., Cheng, M.-M., et al. (2022). Re-thinking co-salient object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(8), 4339–4354.
86. Fan, D.-P., Lin, Z., Ji, G.-P., Zhang, D., Fu, H., & Cheng, M.-M. (2020). Taking a deeper look at co-salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2916–2926). Los Alamitos: IEEE.
87. Zhang, D., Fu, H., Han, J., Borji, A., & Li, X. (2018). A review of co-saliency detection algorithms: fundamentals, applications, and challenges. *ACM Transactions on Intelligent Systems and Technology*, *9*(4), 1–31.
88. Borji, A., Cheng, M.-M., Hou, Q., Jiang, H., & Li, J. (2019). Salient object detection: a survey. *Computational Visual Media*, *5*(2), 117–150.
89. Wang, W., Lai, Q., Fu, H., Shen, J., Ling, H., & Yang, R. (2021). Salient object detection in the deep learning era: an in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 3239–3259.
90. Borji, A., Cheng, M.-M., Jiang, H., & Li, J. (2015). Salient object detection: a benchmark. *IEEE Transactions on Image Processing*, *24*(12), 5706–5722.
91. Zhou, T., Fan, D.-P., Cheng, M.-M., Shen, J., & Shao, L. (2021). RGB-D salient object detection: a survey. *Computational Visual Media*, *7*(1), 37–69.
92. Fan, D.-P., Lin, Z., Zhang, Z., Zhu, M., & Cheng, M.-M. (2020). Rethinking RGB-D salient object detection: models, data sets, and large-scale benchmarks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(5), 2075–2089.
93. Cong, R., Zhang, K., Zhang, C., Zheng, F., Zhao, Y., Huang, Q., et al. (2022). Does thermal really always matter for RGB-T salient object detection? *IEEE Transactions on Multimedia*. Advance online publication. https://doi.org/10.1109/TMM.2022.3216476
94. Tu, Z., Li, Z., Li, C., Lang, Y., & Tang, J. (2021). Multi-interactive dual-decoder for RGB-thermal salient object detection. *IEEE Transactions on Image Processing*, *30*, 5678–5691.
95. Fu, K., Jiang, Y., Ji, G.-P., Zhou, T., Zhao, Q., & Fan, D.-P. (2022). Light field salient object detection: a review and benchmark. *Computational Visual Media*, *8*(4), 509–534.
96. Wang, W., Shen, J., & Shao, L. (2017). Video salient object detection via fully convolutional networks. *IEEE Transactions on Image Processing*, *27*(1), 38–49.
97. Le, T.-N., & Sugimoto, A. (2017). Deeply supervised 3D recurrent FCN for salient object detection in videos. In T. K. Kim, S. Zafeiriou, G. Brostow, et al. (Eds.), *Proceedings of the British machine vision conference* (pp. 1–13). Durham: BMVA Press.
98. Chen, C., Wang, G., Peng, C., Fang, Y., Zhang, D., & Qin, H. (2021). Exploring rich and efficient spatial temporal interactions for real-time video salient object detection. *IEEE Transactions on Image Processing*, *30*, 3995–4007.
99. Le, T.-N., & Sugimoto, A. (2018). Video salient object detection using spatiotemporal deep features. *IEEE Transactions on Image Processing*, *27*(10), 5002–5015.
100. Zhang, M., Liu, J., Wang, Y., Piao, Y., Yao, S., Ji, W., et al. (2021). Dynamic context-sensitive filtering network for video salient object detection. In *2021 IEEE/CVF international conference on computer vision* (pp. 1533–1543). Los Alamitos: IEEE.
101. Li, G., Xie, Y., Wei, T., Wang, K., & Lin, L. (2018). Flow guided recurrent neural encoder for video salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3243–3252). Los Alamitos: IEEE.
102. Song, H., Wang, W., Zhao, S., Shen, J., & Lam, K.-M. (2018). Pyramid dilated deeper convLSTM for video salient object detection. In V. Ferrari, M. Hebert, C. Sminchisescu, et al. (Eds.), *Proceedings of the 15th European conference on computer Vision* (pp. 744–760). Berlin: Springer.
103. Ji, G.-P., Fan, D.-P., Fu, K., Wu, Z., Shen, J., & Shao, L. (2022). Full-duplex strategy for video object segmentation. *Computational Visual Media*, *9*(1), 155–175.

104. Li, H., Chen, G., Li, G., & Yu, Y. (2019). Motion guided attention for video salient object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 7273–7282). Los Alamitos: IEEE.
105. Cong, R., Song, W., Lei, J., Yue, G., Zhao, Y., & Psnet, S. K. (2023). Parallel symmetric network for video salient object detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *7*(2), 402–414.
106. Fan, D.-P., Wang, W., Cheng, M.-M., & Shen, J. (2019). Shifting more attention to video salient object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8554–8564). Los Alamitos: IEEE.
107. Luo, X.-J., Wang, S., Wu, Z., Sakaridis, C., Cheng, Y., Fan, D.-P., et al. (2023). *CamDiff*: *camouflage image augmentation via diffusion*. arXiv preprint arXiv:2304.05469.
108. Li, A., Zhang, J., Lv, Y., Liu, B., Zhang, T., & Dai, Y. (2021). Uncertainty-aware joint salient object and camouflaged object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10071–10081). Los Alamitos: IEEE.
109. Qin, X., Dai, H., Hu, X., Fan, D.-P., Shao, L., & Van Gool, L. (2022). Highly accurate dichotomous image segmentation. In S. Avidan, G. J. Brostow, M. Cissé, et al. (Eds.), *Proceedings of the 17th European conference of computer vision* (pp. 38–56). Berlin: Springer.
110. Yan, J., Le, T.-N., Nguyen, K.-D., Tran, M.-T., Do, T.-T., & Nguyen, T. V. (2021). MirrorNet: bio-inspired camouflaged object segmentation. *IEEE Access*, *9*, 43290–43300.
111. Xiang, M., Zhang, J., Lv, Y., Li, A., Zhong, Y., & Dai, Y. (2021). *Exploring depth contribution for camouflaged object detection*. arXiv preprint arXiv:2106.13217.
112. Wang, K., Bi, H., Zhang, Y., Zhang, C., Liu, Z., & Zheng, S. (2022). D^2c-net: a dual-branch, dual-guidance and cross-refine network for camouflaged object detection. *IEEE Transactions on Industrial Electronics*, *69*(5), 5364–5374.
113. Sun, Y., Chen, G., Zhou, T., Zhang, Y., & Liu, N. (2021). Context-aware cross-level fusion network for camouflaged object detection. In Z.-H. Zhou (Ed.), *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 1025–1031). IJCAI.
114. Kajiura, N., Liu, H., & Satoh, S. (2021). Improving camouflaged object detection with the uncertainty of pseudo-edge labels. In C. Chen, H. Huang, J. Zhou, et al. (Eds.), *ACM multimedia Asia* (pp. 1–7). New York: ACM.
115. Zhu, J., Zhang, X., Zhang, S., & Liu, J. (2021). Inferring camouflaged objects by texture-aware interactive guidance network. In *Proceedings of the 35th AAAI conference on artificial intelligence* (pp. 3599–3607). Menlo Park: AAAI Press.
116. Zhai, Q., Li, X., Yang, F., Chen, C., Cheng, H., & Fan, D.-P. (2021). Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 12997–13007). Los Alamitos: IEEE.
117. Yang, F., Zhai, Q., Li, X., Huang, R., Luo, A., Cheng, H., et al. (2021). Uncertainty-guided transformer reasoning for camouflaged object detection. In *2021 IEEE/CVF international conference on computer vision* (pp. 4126–4135). Los Alamitos: IEEE.
118. Qin, X., Fan, D.-P., Huang, C., Diagne, C., Zhang, Z., Sant'Anna, A. C., et al. (2022). *Boundary-aware segmentation network for mobile and web applications*. arXiv preprint arXiv:2101.04704.
119. Zhang, C., Wang, K., Bi, H., Liu, Z., & Yang, L. (2022). Camouflaged object detection via neighbor connection and hierarchical information transfer. *Computer Vision and Image Understanding*, *221*, 103450.
120. Zhai, W., Cao, Y., Xie, H., & Zha, Z.-J. (2022). Deep texton-coherence network for camouflaged object detection. *IEEE Transactions on Multimedia*. Advance online publication. https://doi.org/10.1109/TMM.2022.3188401
121. Chen, G., Liu, S.-J., Sun, Y.-J., Ji, G.-P., Wu, Y.-F., & Zhou, T. (2022). Camouflaged object detection via context-aware cross-level fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(10), 6981–6993.
122. Zhuge, M., Lu, X., Guo, Y., Cai, Z., & Chen, S. (2022). Cubenet: X-shape connection for camouflaged object detection. *Pattern Recognition*, *127*, 108644.
123. Ji, G.-P., Zhu, L., Zhuge, M., & Fu, K. (2022). Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognition*, *123*, 108414.
124. Zhang, Q., Ge, Y., Zhang, C., & Bi, H. (2022). TprNet: camouflaged object detection via transformer-induced progressive refinement network. *The Visual Computer*. Advance online publication. https://doi.org/10.1007/s00371-022-02611-1
125. Cheng, Y., Hao, H.-Z., Ji, Y., Li, Y., & Liu, C.-P. (2022). Attention-based neighbor selective aggregation network for camouflaged object detection. In *International joint conference on neural networks* (pp. 1–8). Los Alamitos: IEEE.
126. Zhu, H., Li, P., Xie, H., Yan, X., Liang, D., Chen, D., et al. (2022). I can find you! Boundary-guided separated attention network for camouflaged object detection. In *Proceedings of the 36th AAAI conference on artificial intelligence* (pp. 3608–3616). Menlo Park: AAAI Press.
127. Zhou, T., Zhou, Y., Gong, C., Yang, J., & Zhang, Y. (2022). Feature aggregation and propagation network for camouflaged object detection. *IEEE Transactions on Image Processing*, *31*, 7036–7047.
128. Li, P., Yan, X., Zhu, H., Wei, M., Zhang, X.-P., & Findnet, J. Q. (2022). Can you find me? Boundary-and-texture enhancement network for camouflaged object detection. *IEEE Transactions on Image Processing*, *31*, 6396–6411.
129. Chou, M.-C., Chen, H.-J., & Shuai, H.-H. (2022). Finding the Achilles heel: progressive identification network for camouflaged object detection. In *IEEE international conference on multimedia and expo* (pp. 1–6). Los Alamitos: IEEE.
130. Liu, J., Zhang, J., & Barnes, N. (2022). Modeling aleatoric uncertainty for camouflaged object detection. In *IEEE/CVF winter conference on applications of computer vision* (pp. 2613–2622). Los Alamitos: IEEE.
131. Sun, Y., Wang, S., Chen, C., & Xiang, T.-Z. (2022). Boundary-guided camouflaged object detection. In L.de. Raedt (Ed.), *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 1335–1341). IJCAI.
132. Zhang, M., Xu, S., Piao, Y., Shi, D., Lin, S., & Lu, H. (2022). PreyNet: preying on camouflaged objects. In J. Magalhães, A. del Bimbo, S. Satoh, et al. (Eds.), *The 30th ACM international conference on multimedia* (pp. 5323–5332). New York: ACM.
133. Liu, Z., Zhang, Z., Tan, Y., & Wu, W. (2022). Boosting camouflaged object detection with dual-task interactive transformer. In *Proceedings of the 26th international conference on pattern recognition* (pp. 140–146). Los Alamitos: IEEE.
134. Pang, Y., Zhao, X., Xiang, T.-Z., Zhang, L., & Lu, H. (2022). Zoom in and out: a mixed-scale triplet network for camouflaged object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2150–2160). Los Alamitos: IEEE.
135. Zhong, Y., Li, B., Tang, L., Kuang, S., Wu, S., & Ding, S. (2022). Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4494–4503). Los Alamitos: IEEE.
136. Jia, Q., Yao, S., Liu, Y., Fan, X., Liu, R., & Luo, Z. (2022). Segment, magnify and reiterate: detecting camouflaged objects the hard way. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4703–4712). Los Alamitos: IEEE.
137. Zhai, Q., Li, X., Yang, F., Jiao, Z., Luo, P., Cheng, H., et al. (2023). MGL: mutual graph learning for camouflaged object detection. *IEEE Transactions on Image Processing*, *32*, 1897–1910.
138. Lin, J., Tan, X., Xu, K., Ma, L., & Lau, R. W. (2023). Frequency-aware camouflaged object detection. *ACM Transactions on Multimedia Computing Communications and Applications*, *19*(2), 1–16.
139. Ren, J., Hu, X., Zhu, L., Xu, X., Xu, Y., Wang, W., et al. (2023). Deep texture-aware features for camouflaged object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(3), 1157–1167.
140. Xing, H., Wang, Y., Wei, X., Tang, H., Gao, S., & Zhang, W. (2023). Go closer to see better: camouflaged object detection via object area amplification and figure-ground conversion. *IEEE Transactions on Circuits and Systems for Video Technology*. Advance online publication. https://doi.org/10.1109/TCSVT.2023.3255304
141. Zheng, D., Zheng, X., Yang, L. T., Gao, Y., Zhu, C., & Mffn, Y. R. (2023). Multi-view feature fusion network for camouflaged object detection. In *IEEE/CVF winter conference on applications of computer vision* (pp. 6221–6231). Los Alamitos: IEEE.
142. He, R., Dong, Q., Lin, J., & Lau, R. W. (2023). Weakly-supervised camouflaged object detection with scribble annotations. In B. Williams, Y. Chen, & J. Neville (Eds.), *Proceedings of the 37th AAAI conference on artificial intelligence* (pp. 781–789). Menlo Park: AAAI Press.

143. Hu, X., Fan, D.-P., Qin, X., Dai, H., Ren, W., Tai, Y., et al. (2023). High-resolution iterative feedback network for camouflaged object detection. In B. Williams, Y. Chen, & J. Neville (Eds.), *Proceedings of the 37th AAAI conference on artificial intelligence* (pp. 881–889). Menlo Park: AAAI Press.
144. Huang, Z., Dai, H., Xiang, T.-Z., Wang, S., Chen, H.-X., Qin, J., et al. (2023). Feature shrinkage pyramid for camouflaged object detection with transformers. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5557–5566). Los Alamitos: IEEE.
145. He, C., Li, K., Zhang, Y., Tang, L., Zhang, Y., Guo, Z., et al. (2023). Camouflaged object detection with feature decomposition and edge reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 22046–22055). Los Alamitos: IEEE.
146. Luo, N., Pan, Y., Sun, R., Zhang, T., Xiong, Z., & Wu, F. (2023). Camouflaged instance segmentation via explicit de-camouflaging. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 17918–17927). Los Alamitos: IEEE.
147. Sun, W., Liu, C., Zhang, L., Li, Y., Wei, P., Liu, C., et al. (2022). *Dqnet: cross-model detail querying for camouflaged object detection*. arXiv preprint arXiv:2212.08296.
148. Yin, B. Zhang, X., Hou, Q., Sun, B.-Y., Fan, D.-P., & van Gool, L. (2023). *Camoformer: masked separable attention for camouflaged object detection*. arXiv preprint arXiv:2212.06570.
149. Wu, Z., Paudel, D. P., Fan, D.-P., Wang, J., Wang, S., Demonceaux, C., et al. (2023). *Source-free depth for object pop-out*. arXiv preprint arXiv:2212.05370.
150. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440). Los Alamitos: IEEE.
151. Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *2015 IEEE international conference on computer vision* (pp. 1395–1403). Los Alamitos: IEEE.
152. Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. (2015). Deeply-supervised nets. In *Proceedings of the 18th international conference on artificial intelligence and statistics* (pp. 562–570). JMLR.
153. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 936–944). Los Alamitos: IEEE.
154. Xie, C., Xiang, Y., Harchaoui, Z., & Fox, D. (2019). Object discovery in videos as foreground motion clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9994–10003). Los Alamitos: IEEE.
155. Lamdouar, H., Xie, W., & Zisserman, A. (2021). Segmenting invisible moving objects. In *Proceedings of the British machine vision conference* (pp. 1–14). Durham: BMVA Press.
156. Yang, C., Lamdouar, H., Lu, E., Zisserman, A., & Xie, W. (2021). Self-supervised video object segmentation by motion grouping. In *2021 IEEE/CVF international conference on computer vision* (pp. 7157–7168). Los Alamitos: IEEE.
157. Bideau, P., Learned-Miller, E., Schmid, C., & Alahari, K. (2022). *The right spin: learning object motion from rotation-compensated flow fields*. arXiv preprint arXiv:2203.00115.
158. Xie, J., Xie, W., & Zisserman, A. (2022). Segmenting moving objects via an object-centric layered representation. In S. Koyejo, S. Mohamed, A. Agarwal, et al. (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 1–14). Red Hook: Curran Associates.
159. Meunier, E., Badoual, A., & Bouthemy, P. (2023). EM-driven unsupervised learning for efficient motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(4), 4462–4473.
160. Kowal, M., Siam, M., Islam, M. A., Bruce, N. D. B., Wildes, R. P., & Derpanis, K. G. (2022). A deeper dive into what deep spatiotemporal networks encode: quantifying static vs. dynamic information. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 13979–13989). Los Alamitos: IEEE.
161. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., & Sorkine-Hornung, A. (2016). A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 724–732). Los Alamitos: IEEE.
162. Ochs, P., Malik, J., & Brox, T. (2013). Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*(6), 1187–1200.
163. Bideau, P., & Learned-Miller, E. (2016). It's moving! A probabilistic model for causal motion segmentation in moving camera videos. In B. Leibe, J. Matas, N. Sebe, et al. (Eds.), *Proceedings of the 14th European conference on computer Vision* (pp. 433–449). Berlin: Springer.
164. Li, L., Zhou, T., Wang, W., Yang, L., Li, J., & Yang, Y. (2022). Locality-aware inter-and intra-video reconstruction for self-supervised correspondence learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8709–8720). Los Alamitos: IEEE.
165. Araslanov, N., Schaub-Meyer, S., & Roth, S. (2021). Dense unsupervised learning for video segmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, et al. (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 25308–25319). Red Hook: Curran Associates.
166. Liu, R., Wu, Z., Yu, S., & Lin, S. (2021). The emergence of objectness: learning zero-shot segmentation from videos. In M. Ranzato, A. Beygelzimer, Y. Dauphin, et al. (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 13137–13152). Red Hook: Curran Associates.
167. Lu, X., Wang, W., Shen, J., Tai, Y.-W., Crandall, D. J., & Hoi, S. C. (2020). Learning video object segmentation from unlabeled videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8957–8967). Los Alamitos: IEEE.
168. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision* (pp. 9630–9640). Los Alamitos: IEEE.
169. Wang, Z., Zhao, H., Li, Y.-L., Wang, S., Torr, P., & Bertinetto, L. (2021). Do different tracking tasks require different appearance models? In M. Ranzato, A. Beygelzimer, Y. Dauphin, et al. (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 726–738). Red Hook: Curran Associates.
170. Yan, B., Jiang, Y., Sun, P., Wang, D., Yuan, Z., Luo, P., et al. (2022). Towards grand unification of object tracking. In S. Avidan, G. J. Brostow, M. Cissé, et al. (Eds.), *Proceedings of the 17th European conference of computer vision* (pp. 733–751). Berlin: Springer.
171. Xu, H., Zhang, J., Cai, J., Rezatofighi, H., Yu, F., Tao, D., et al. (2022). *Unifying flow, stereo and depth estimation*. arXiv preprint arXiv:2211.05783.
172. Teed, Z., & Deng, J. (2020). Raft: recurrent all-pairs field transforms for optical flow. In A. Vedaldi, H. Bischof, T. Brox, et al. (Eds.), *Proceedings of the 15th European conference on computer Vision* (pp. 402–419). Berlin: Springer.
173. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). Los Alamitos: IEEE.
174. Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., & Torr, P. (2019). Res2net: a new multi-scale backbone architecture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(2), 652–662.
175. Tan, M., & Le, Q. (2019). Efficientnet: rethinking model scaling for convolutional neural networks. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (pp. 6105–6114). PMLR.
176. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 11966–11976). Los Alamitos: IEEE.
177. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Segformer, P. L. (2021). Simple and efficient design for semantic segmentation with transformers. In M. Ranzato, A. Beygelzimer, Y. Dauphin, et al. (Eds.), *Advances in neural information processing systems* (Vol. 34, pp. 12077–12090). Red Hook: Curran Associates.
178. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., et al. (2022). PVT v2: improved baselines with pyramid vision transformer. *Computational Visual Media*, *8*(3), 415–424.
179. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision* (pp. 9992–10002). Los Alamitos: IEEE.
180. Fan, D.-P., Ji, G.-P., Qin, X., & Cheng, M.-M. (2021). Cognitive vision inspired object segmentation metric and loss function. *Scientia Sinica Informationis*, *51*(9), 1475–1489.
181. Fan, D.-P., Gong, C., Cao, Y., Ren, B., Cheng, M.-M., & Borji, A. (2018). Enhanced-alignment measure for binary foreground map evaluation. In J. Lang (Ed.), *Proceedings of the 31st international joint conference on artificial intelligence* (pp. 698–704). IJCAI.

182. Fan, D.-P., Cheng, M.-M., Liu, Y., Li, T., & Borji, A. (2017). Structure-measure: a new way to evaluate foreground maps. In *2017 IEEE international conference on computer vision* (pp. 4558–4567). Los Alamitos: IEEE.
183. Cheng, M.-M., & Fan, D.-P. (2021). Structure-measure: a new way to evaluate foreground maps. *International Journal of Computer Vision*, *129*(9), 2622–2638.
184. Qi, J., Gao, Y., Hu, Y., Wang, X., Liu, X., Bai, X., et al. (2022). Occluded video instance segmentation: a benchmark. *International Journal of Computer Vision*, *130*(8), 2022–2039.
185. Wang, M., & Deng, W. (2018). Deep visual domain adaptation: a survey. *Neurocomputing*, *312*, 135–153.
186. Yin, N., Shen, L., Wang, M., Lan, L., Ma, Z., Chen, C., et al. (2023). CoCo: a coupled contrastive framework for unsupervised domain adaptive graph classification. In *Proceedings of the 40th international conference on machine learning* (pp. 1–14). PMLR.
187. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.
188. Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: a survey on few-shot learning. *ACM Computing Surveys*, *53*(3), 1–34.
189. Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5149–5169.
190. Cao, Y., Li, S., Liu, Y., Yan, Z., Dai, Y., Yu, P. S., et al. (2023). *A comprehensive survey of ai-generated content (aigc): a history of generative ai from GAN to ChatGPT*. arXiv preprint arXiv:2303.04226.
191. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.
192. Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In [Poster presentation]. *Proceedings of the 4th international conference on learning representations, San Juan, Puerto Rico*.
193. Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *Proceedings of the 6th international conference on learning representations* (pp. 1–26). ICLR.
194. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10674–10685). Los Alamitos: IEEE.
195. Zhang, L., & Agrawala, M. (2023). *Adding conditional control to text-to-image diffusion models*. arXiv preprint arXiv:2302.05543.
196. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., et al. (2023). *Segment anything*. arXiv preprint arXiv:2304.02643.
197. Ji, G.-P., Fan, D.-P., Xu, P., Cheng, M.-M., Zhou, B., & Van Gool, L. (2023). *Sam struggles in concealed scenes–empirical study on "segment anything"*. arXiv preprint arXiv:2304.06022.
198. Masci, J., Meier, U., Ciresan, D., Schmidhuber, J., & Fricout, G. (2012). Steel defect classification with max-pooling convolutional neural networks. In *Proceedings of the 2012 international joint conference on neural networks* (pp. 1–6). Los Alamitos: IEEE.
199. Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, *53*(6), 1517–1525.
200. Luo, Q., Fang, X., Su, J., Zhou, J., Zhou, B., Yang, C., et al. (2020). Automated visual defect classification for flat steel surface: a survey. *IEEE Transactions on Instrumentation and Measurement*, *69*(12), 9329–9349.
201. Ngan, H. Y., Pang, G. K., & Yung, N. H. C. (2011). Automated fabric defect detection—a review. *Image and Vision Computing*, *29*(7), 442–458.
202. Kumar, A. (2008). Computer-vision-based fabric defect detection: a survey. *IEEE Transactions on Industrial Electronics*, *55*(1), 348–363.
203. Ghorai, S., Mukherjee, A., Gangadaran, M., & Dutta, P. K. (2012). Automatic defect detection on hot-rolled flat steel products. *IEEE Transactions on Instrumentation and Measurement*, *62*(3), 612–621.
204. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., & Steger, C. (2018). Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In *Proceedings of the 14th international joint conference on computer vision, imaging and computer graphics theory and applications* (pp. 372–380). Setúbal: SciTePress.
205. Tabernik, D., Šela, S., Skvarč, J., & Skočaj, D. (2020). Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, *31*(3), 759–776.
206. Tsai, D.-M., Fan, S.-K. S., & Chou, Y.-H. (2021). Auto-annotated deep segmentation for surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–10.
207. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., & Steger, C. (2021). The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, *129*(4), 1038–1059.
208. Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019). MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9592–9600). Los Alamitos: IEEE.
209. Song, K.-C., Hu, S.-P., Yan, Y.-H., & Li, J. (2014). Surface defect detection method using saliency linear scanning morphology for silicon steel strip under oil pollution interference. *ISIJ International*, *54*(11), 2598–2607.
210. Bao, Y., Song, K., Liu, J., Wang, Y., Yan, Y., Yu, H., et al. (2021). Triplet-graph reasoning network for few-shot metal generic surface defect segmentation. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–11.
211. He, Y., Song, K., Meng, Q., & Yan, Y. (2019). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Transactions on Instrumentation and Measurement*, *69*(4), 1493–1504.
212. Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, *17*(12), 3434–3445.
213. Cui, L., Qi, Z., Chen, Z., Meng, F., & Shi, Y. (2015). Pavement distress detection using random decision forests. In C. Zhang, W. Huang, Y. Shi, et al. (Eds.), *Proceedings of the 2nd international conference on data science* (pp. 95–102). Berlin: Springer.
214. Huang, Y., Qiu, C., & Yuan, K. (2020). Surface defect saliency of magnetic tile. *The Visual Computer*, *36*(1), 85–96.
215. Ji, G.-P., Zhuge, M., Gao, D., Fan, D.-P., Sakaridis, C., & Van Gool, L. (2023). Masked vision-language transformer in fashion. *Machine Intelligence Research*, *20*(3), 421–434.

Publisher's Note