Machine Intelligence Research
www.mi-research.net

20(1), February 2023, 92-108
DOI: 10.1007/s11633-022-1365-9

# Deep Gradient Learning for Efficient Camouflaged Object Detection

Ge-Peng Ji[1] Deng-Ping Fan[2*] Yu-Cheng Chou[1] Dengxin Dai[2]
Alexander Liniger[2] Luc Van Gool[2]

[1] School of Computer Science, Wuhan University, Wuhan 430072, China

[2] Computer Vision Laboratory, ETH Zürich, Zürich 8092, Switzerland

**Abstract:** This paper introduces deep gradient network (DGNet), a novel deep framework that exploits object gradient supervision for camouflaged object detection (COD). It decouples the task into two connected branches, i.e., a context and a texture encoder. The essential connection is the gradient-induced transition, representing a soft grouping between context and texture features. Benefiting from the simple but efficient framework, DGNet outperforms existing state-of-the-art COD models by a large margin. Notably, our efficient version, DGNet-S, runs in real-time (80 fps) and achieves comparable results to the cutting-edge model JCSOD-CVPR$_{21}$ with only 6.82% parameters. The application results also show that the proposed DGNet performs well in the polyp segmentation, defect detection, and transparent object segmentation tasks. The code will be made available at https://github.com/GewelsJI/DGNet.

**Keywords:** Camouflaged object detection (COD), object gradient, soft grouping, efficient model, image segmentation.

**Citation:** G. P. Ji, D. P. Fan, Y. C. Chou, D. Dai, A. Liniger, L. V. Gool. Deep gradient learning for efficient camouflaged object detection. *Machine Intelligence Research*, vol.20, no.1, pp.92–108, 2023. http://doi.org/10.1007/s11633-022-1365-9

## 1 Introduction

Camouflaged object detection[1, 2] (COD) aims to segment objects with either artificial or natural patterns where objects "perfectly" blend into the background to avoid being discovered[2]. Several successful applications, such as medical image analysis (e.g., polyp[3–5] and lung infection[6–8] segmentation), video analysis (e.g., motion segmentation[9], surveillance[10], and autonomous driving[11]), and recreational art[12, 13], have shown COD's scientific and practical value.

Recent studies[14–17] present compelling results based on the supervision of the whole object-level ground-truth mask. Later, various sophisticated techniques, e.g., boundary-based[18–20] and uncertainty-guided[21, 22], were developed to augment COD's underlying representations. However, features learned from boundary-supervised or uncertainty-based models usually respond to the sparse edge of camouflage objects, thereby introducing noisy features, especially for complex scenes (see Fig. 1(a)). Besides, the boundaries of camouflaged objects are always "indefinable" or "fuzzy"; thus, they do not be pop-out from a quick visual scanning. We notice that despite the object's camouflage, there are still some clues left, shown in the first column of Fig. 1 (white speckles). Instead of extracting only boundary or uncertainty regions, we are interested in how the network mines these "discriminative patterns" inside the object.

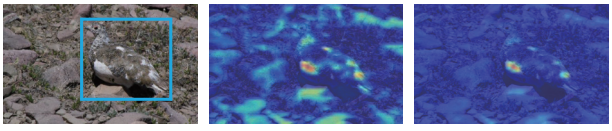


Image (a) Object boundary (b) Object gradient

Fig. 1 Feature visualization of learned texture. We observe that the proposed DGNet-S under the object boundary supervision (a) contains diffused noises in the background. By contrast, object gradient supervision (b) enforces the network focus on the regions where the intensity changes dramatically.

From this perspective, we present our deep gradient network (DGNet) via the explicit supervision of the object-level gradient map. The underlying hypothesis is that there are some intensity changes inside the camouflaged objects. To ease the learning task, we decouple the DGNet into two connected branches, i.e., a context and a texture encoder. The former can be viewed as a contextual semantics learner, while the latter acts as a structural texture extractor. In this way, we can alleviate the feature ambiguity between the high-level and low-level features extracted from the individual branch. To suffi-

Research Article
Manuscript received May 25, 2022; accepted August 6, 2022
Recommended by Associate Editor Alan Wee-Chung Liew
*Corresponding author (dengpfan@gmail.com). The major part of this work was done while Ge-Peng Ji was an intern mentored by Deng-Ping Fan.

ciently aggregate the above two discriminative features generated by the two branches, we further design a gradient-induced transition (GIT) module that collaboratively ensembles the multi-source feature space at different group scales (i.e., soft grouping). Fig. 1(b) shows that our DGNet can detect texture patterns while suppressing the background noise by an intensity-sensitive strategy focusing on the intra-region of a camouflaged object.

Extensive experiments on three challenging COD benchmarks illustrate that the proposed DGNet achieves state-of-the-art (SOTA) performance without introducing any complicated structures. Furthermore, we implement an efficient version, DGNet-S, with 8.3 M parameters, which achieves the fastest inference speed (80 fps) among COD-related baselines. Notably, it only has 6.82% parameters compared to the cutting-edge model JCSOD-CVPR$_{21}$[22] while achieving comparable performance. These results show that DGNet significantly narrows the gap between scientific research and practical application. Three downstream applications (see Section 5) of our DGNet also support this conclusion. The major contributions of this paper are summarized as follows:

1) We introduce a novel deep gradient-based framework, dubbed DGNet, for addressing the camouflaged object detection task.

2) We propose a gradient-induced transition to automatically group features from the context and texture branches according to the soft grouping strategy.

3) We present three applications and achieve good performance, including polyp segmentation, defect detection, and transparent object segmentation.

## 2 Prior works

Traditional methods detect camouflaged objects via extracting various hand-crafted features between the camouflaged areas and their backgrounds, which calculate the 3D convexity[23], co-occurrence matrix[24], expectation-maximization statistics[25], optical flow[26], and Gaussian mixture model[27]. These methods work well for simple backgrounds, while the performance degrades drastically for complex backgrounds.

CNN-based approaches could be generally categorized into three strategies: 1) Attention-based strategy: Sun et al.[28, 29] introduce a network with an attention-induced cross-level fusion module to integrate multi-scale features and a dual-branch global context module to mine multi-scale contextual information. To mimic the detection process of predators, Mei et al.[14] develop PFNet, which contains a positioning and focusing module to conduct the identification. Some works propose delicate structures such as covariance matrices of feature[30] and multivariate calibration components[19] to improve the robustness of the network. Kajiura et al.[31] improve the detection accuracy by exploring the uncertainties of pseudo-edge and pseudo-map labels. Zhuge et al.[32] propose a cube-alike architecture for COD, which accompanies attention fusion and X-shaped connection to integrate multiple-layer features sufficiently. 2) Two-stage strategy: Search and identification strategy[1] is an early practice to model the COD task. In [2], the neighbor connection decoder and group-reversal attention are introduced in SINet[1] to boost the performance further. 3) Joint-learning strategy: ANet[33] is an early attempt to utilize the classification and segmentation scheme for COD. LSR[15] and JCSOD[22] have recently renewed the joint-learning framework by introducing camouflaged ranking or learning from salient objects to camouflaged objects. ZoomNet[34] is a mixed-scale triplet network that employs the zoom strategy to learn the discriminative camouflaged semantics.

Transformer-based & Graph-based models are two recent technology trends. Recently, Mao et al.[35] introduced the concept of difficulty-aware learning based on the Transformer for both camouflaged and salient object detection. UGTR[21] explicitly utilized the probabilistic representational model to learn the uncertainties of the camouflaged object under the Transformer framework. In addition, Cheng et al.[36] are the first to collect a video dataset for COD and utilize the Transformer-based framework to exploit short-term dynamics and long-term temporal consistency to detect dynamic camouflaged objects. Later, Zhai et al.[18] designed the mutual graph learning model, which decouples one input into different features for roughly locating the target and accurately capturing its boundary.

**Remarks.** On the contrary, our work excavates the texture information by learning the object-level gradient rather than using boundary-aware or uncertainty-aware modelling. The biologically inspired idea behind this is that the abundant gradient cues inside the camouflaged object deserve to be explored, while the sparse boundary cues are insufficient to achieve this. As shown in Fig. 2, we also note that the recent work[37] tries to utilize the texture cues while they discard excessive object gradient cues due to different threshold settings of the Canny detector. In short, this paper aims to design an elegant framework towards efficient COD with more concise ideas (i.e., object gradient learning). More experimental validations are discussed in Section 4.3.

## 3 Deep gradient network

As discussed in [38], the low-level and high-level features occupy an equal role in the scene understanding. As suggested by [39], it is not encouraged to encode them simultaneously. As shown in Fig. 3, we propose to model the camouflaged representations with two separate encoders, a context and a texture encoder.

### 3.1 Context encoder

For a camouflaged input image $\mathcal{I} \in \mathbf{R}^{3\times H\times W}$, we use the widely used EfficientNet[40] as the context encoder to

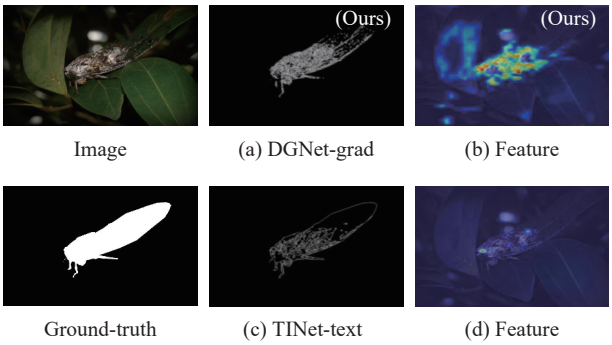



Fig. 2 Compared to the texture label proposed in TINet[37], our object gradient label (a) keeps more geometric cues inside the camouflaged object. DGNet, under the supervision of texture label (c), fails to infer attentive regions (d) since the imbalanced distribution of sparse pixels (e.g., thin object boundaries). Notably, such improvement exerts our DGNet more robust with the reliable auxiliaries, e.g., feature in (b).

obtain the pyramid features $\{\boldsymbol{X}_i\}_{i=1}^{5}$.

**Dimensional reduction.** Inspired by [3], we adopt the following two steps to ensure efficient element-wise operations between different levels in the decoding stage: 1) We only pick out the top-three features (i.e., when $i = 3, 4, 5$), which retain the affluent semantics of the visual scene. 2) We further utilize two stacked ConvBR[1] layers with $C_i \times 3 \times 3$ filters to reduce the dimension of each candidate feature to $C_i$, contributing to easing the computational burden of subsequent operations. The final outputs are three context features $\{\boldsymbol{X}_i^R\}_{i=3}^{5} \in \mathbf{R}^{C_i \times H_i \times W_i}$, where $C_i$, $H_i = H/2^i$, and $W_i = W/2^i$ denote the channel, height, and width of the feature maps.

## 3.2 Texture encoder

We also introduce a tailored texture branch supervised by the object-level gradient map, compensating for the pattern degradation caused by the top-three context features′ weak representation of geometric textures.

**Object gradient generation.** An image gradient describes the directional change in an image′s intensity or color between adjacent positions, which is widely applied for edge detection[43] and super-resolution[44]. The right part of Fig. 3 presents four widely used types of supervision labels. The object boundary (c) and image gradient (e) can be directly generated by calculating the gradient of the object-level ground-truth $\boldsymbol{Z}^C$ (b) and raw image (a), respectively. However, the raw image gradient map (e), which contains irrelevant background noises, may mislead the optimization process when serving as the supervision signal for texture learning. To address this problem, we introduce a novel camouflage learning paradigm that uses the object-level gradient map $\boldsymbol{Z}^G$ (d) as supervision, which holds both the gradient cues of the object′s boundaries and interior regions. This process could be formulated as

$$\boldsymbol{Z}^G = \mathcal{F}_E(\mathcal{I}(x, y)) \circledast \boldsymbol{Z}^C \tag{1}$$

where $\mathcal{F}_E$ represents the standard Canny edge detector[45] for input $\mathcal{I}$ with discrete pixel coordinates $(x, y)$. $\circledast$ means the element-wise multiplication.

**Texture encoder.** Because low-level features with a high resolution will introduce a computational burden, we design a tailored lightweight encoder instead of utilizing an out-of-box backbone. We obtain the texture feature $\boldsymbol{X}^G \in \mathbf{R}^{C_g \times H_g \times W_g}$ from layer#03 (see Table 1). However, we supervise the following layer#04 with the object-level gradient $\boldsymbol{Z}^G$. We keep the texture feature with a larger resolution (i.e., $H_g = H/8$ and $W_g = W/8$) since the features with smaller resolution would discard most geometric details.

## 3.3 Gradient-induced transition

The latent correlation between context and texture features offers great potential for adaptive fusion rather than adopting naive fusion strategies (e.g., concatenation and addition operations). Here, we design a flexible plug-and-play gradient-induced transition (GIT) module (see Fig. 4), which views the texture feature as the auxiliaries in the multi-source aggregation from a group-wise perspective. Specifically, it comprises the following three steps.

**Gradient-induced group learning.** Inspired by [2], we first adopt the gradient-induced group learning strategy, which splits three context features $\{\boldsymbol{X}_i^R\}_{i=3}^{5}$ and a texture feature $\boldsymbol{X}^G$ into fixed groups along the channel dimension. This strategy can be formulated for each $\boldsymbol{X}_i^R$ and $\boldsymbol{X}^G$ pair:

$$\begin{aligned} \{\boldsymbol{X}_{i,m}^R\}_{m=1}^{M} \in \mathbf{R}^{K_i \times H_i \times W_i} &\leftarrow \boldsymbol{X}_i^R \in \mathbf{R}^{C_i \times H_i \times W_i} \\ \{\boldsymbol{X}_m^G\}_{m=1}^{M} \in \mathbf{R}^{K_g \times H_g \times W_g} &\leftarrow \boldsymbol{X}^G \in \mathbf{R}^{C_g \times H_g \times W_g} \end{aligned} \tag{2}$$

where $\leftarrow$ is the feature grouping operation. $K_i = C_i/M$ and $K_g = C_g/M$ denote the channel number of each feature group, and $M$ is the corresponding number of groups. Then, we periodically arrange groups of context features $\boldsymbol{X}_{i,m}^R$ and texture features $\boldsymbol{X}_m^G$, which generates the regrouped feature $\boldsymbol{Q}_i$ via:

$$\boldsymbol{Q}_i \in \mathbf{R}^{(C_i+C_g) \times H_i \times W_i} = \langle \boldsymbol{Y}_i^1; \cdots; \boldsymbol{Y}_i^m; \cdots; \boldsymbol{Y}_i^M \rangle \tag{3}$$

where $\langle \cdot; \cdot \rangle$ means the channel-wise feature concatenation. Here, the $m$-th sub-component $\boldsymbol{Y}_i^m$ is derived from

$$\boldsymbol{Y}_i^m \in \mathbf{R}^{(K_i+K_g) \times H_i \times W_i} = \langle \mathcal{F}_{\downarrow}(\boldsymbol{X}_m^G); \boldsymbol{X}_{i,m}^R \rangle \tag{4}$$

where the down-sampling operation $\mathcal{F}_{\downarrow}(\cdot)$ ensures that the

[1] In this paper, ConvBR denotes the standard convolutional layer followed by a BN[41] layer and a ReLU[42] layer.

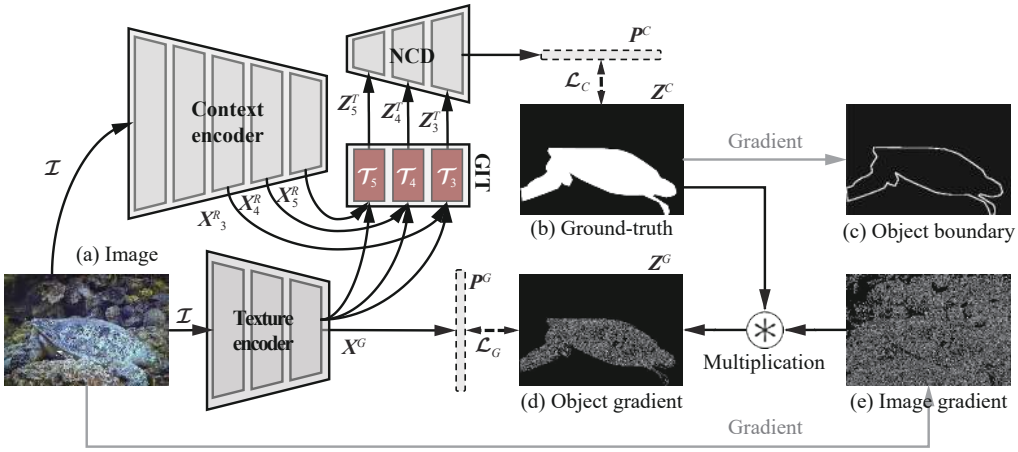



Fig. 3 Overall pipeline of the proposed DGNet. It consists of two connected learning branches, i.e., context encoder (Section 3.1) and texture encoder (Section 3.2). Then, we introduce a gradient-induced transition (GIT) (Section 3.3) to collaboratively aggregate the feature that is derived from the above two encoders. Finally, a neighbor connected decoder (NCD)[2] is adopted to generate the prediction $\boldsymbol{P}^C$ (Section 3.4).

Table 1 Details of the tailored texture encoder. $k$: kernel size, $c$: output channels, $s$: stride, and $p$: zero-padding. Here, we set the channel $C_g = 32$ as the default setting.

| Layer | Input size | Output size | Component | $k$ | $c$ | $s$ | $p$ |
|---|---|---|---|---|---|---|---|
| #01 | $3 \times H \times W$ | $64 \times \frac{H}{2} \times \frac{W}{2}$ | ConvBR | 7 | 64 | 2 | 3 |
| #02 | $64 \times \frac{H}{2} \times \frac{W}{2}$ | $64 \times \frac{H}{4} \times \frac{W}{4}$ | ConvBR | 3 | 64 | 2 | 1 |
| #03 | $64 \times \frac{H}{4} \times \frac{W}{4}$ | $C_g \times \frac{H}{8} \times \frac{W}{8}$ | ConvBR | 3 | $C_g$ | 2 | 1 |
| #04 | $C_g \times \frac{H}{8} \times \frac{W}{8}$ | $1 \times \frac{H}{8} \times \frac{W}{8}$ | ConvBR | 1 | 1 | 1 | 0 |

spatial resolution of $\boldsymbol{X}_m^G$ matches $\boldsymbol{X}_{i,m}^R$.

**Soft grouping strategy.** The naive feature fusion strategies may ignore the correlation or distinctiveness between context and texture representations due to lacking further multi-source interactions. Inspired by the parallel design introduced in [46] for capturing objects at multiple scales, we propose using a soft grouping strategy to provide parallel nonlinear projections at multiple fine-grained sub-spaces, which enables the network to probe multi-source representations jointly. Specifically, we set three parallel sub-branches (i.e., $\{N_1, N_2, N_3\}$ as in the gray region of Fig. 4) for the soft grouping in our experiment. Here, for simplifying illustration, we take the $N$-th sub-branch as an example via omitting the subscript, which could be formulated as

$$\boldsymbol{A}_i^N = \langle \mathcal{F}_N^1(\boldsymbol{Q}_i^1); \cdots; \mathcal{F}_N^n(\boldsymbol{Q}_i^n) \cdots \mathcal{F}_N^N(\boldsymbol{Q}_i^N) \rangle \tag{5}$$

where $\mathcal{F}_N^n(\boldsymbol{Q}_i^n) \in \mathbf{R}^{(C_i/N) \times H_i \times W_i} = f_n(\boldsymbol{Q}_i^n, \omega_n)$ intentionally introduces soft non-linearity at each multi-source sub-space. The projection function $f_n$ is implemented by a convolutional layer with $C_i$ filters of the shape of $(C_i + C_g)/N \times 1 \times 1$, which is parameterized by learnable weights $\omega_n$. Here, $\boldsymbol{Q}_i^n$ is the $n$-th subset of the regrouped feature $\boldsymbol{Q}_i$ that is divided into $N$ groups.

**Parallel residual learning.** We further introduce residual learning[47] in a parallel manner at different group-

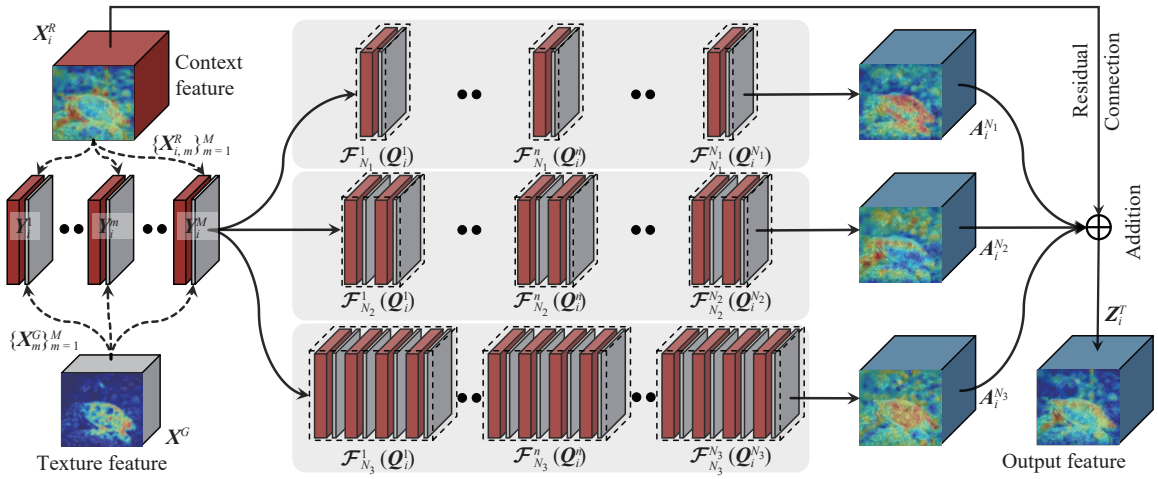



Fig. 4 Illustration of the proposed gradient-induced transition

aware scales. Consequently, we can define the GIT function $\mathcal{T}_i(\cdot,\cdot)$ (see the red block in Fig. 3) as

$$\boldsymbol{Z}_i^{\mathrm{T}} = \mathcal{T}_i(\boldsymbol{X}_i^R, \boldsymbol{X}^G) = \boldsymbol{X}_i^R \oplus \sum_N \boldsymbol{A}_i^N \tag{6}$$

where $N \in \{N_1, N_2, N_3\}$ denotes a set of scaling factors for different groups, which will be discussed in Section 4.3. $\oplus$ means the element-wise addition and $\sum$ denotes a sum of multiple terms. The final output is $\{\boldsymbol{Z}_i^T\}_{i=3}^5 \in \mathbf{R}^{C_i \times H_i \times W_i}$.

### 3.4 Learning details

**Decoder.** Given the context features $\{\boldsymbol{X}_i^R\}_{i=3}^5$, we firstly apply the GIT function $\mathcal{T}_i(\cdot,\cdot)$ (see Equ. (6)) to get the output features $\{\boldsymbol{Z}_i^{\mathrm{T}}\}_{i=3}^5$. To exploit the above gradient-induced features in $\boldsymbol{Z}_i^{\mathrm{T}}$ more efficiently, we utilize the neighbor connection decoder (NCD)[2] to generate the final prediction, enabling feature propagation from high to low levels. Thus, the final prediction $\boldsymbol{P}^C$ can be derived from $\boldsymbol{P}^C \in \mathbf{R}^{1\times H\times W} = \mathrm{NCD}(\boldsymbol{Z}_3^T, \boldsymbol{Z}_4^T, \boldsymbol{Z}_5^T)$.

**Loss function.** The overall optimization objective is defined as

$$\mathcal{L} = \mathcal{L}_C(\boldsymbol{P}^C, \boldsymbol{Z}^C) + \mathcal{L}_G(\boldsymbol{P}^G, \boldsymbol{Z}^G) \tag{7}$$

where $\mathcal{L}_C$ and $\mathcal{L}_G$ represent the segmentation and object gradient loss functions, respectively. For the former, it is formulated as $\mathcal{L}_C = \mathcal{L}_{IoU}^w + \mathcal{L}_{BCE}^w$, where $\mathcal{L}_{IoU}^w$ and $\mathcal{L}_{BCE}^w$ represent the weighted intersection-over-union loss and the weighted binary cross-entropy loss, respectively. They assign the adaptive weight for each pixel according to its difficulty in focusing on the global structure and paying more attention to the hard pixels. The definitions of these losses are the same as in [1, 2, 48], and their effectiveness has been proven in binary segmentation. For the latter, we employ the standard mean squared error loss function.

**Training settings.** The proposed DGNet is implemented in the PyTorch[49]/Jittor[50] toolbox and trained/inferred on a single NVIDIA RTX TITAN GPU. The model parameters are initialized with the strategy of [51], and we initialize the backbone with the model weights pre-trained on ImageNet[52] to prevent over-fitting. We discard the last stage of Conv1×1, pooling, and fully connected layers in the EfficientNet[40] backbone and extract the features from the top-three lateral outputs, including stage-4 ($\boldsymbol{X}_3$), stage-6 ($\boldsymbol{X}_4$), and stage-8 ($\boldsymbol{X}_5$). Considering the performance-efficiency trade-off, we instantiate two variants to adapt the specific requirement under various computational overheads (refer to Table 2).

We train our model end-to-end, using Adam[53]. The cosine annealing part of the SGDR strategy[54] is used to adjust the learning rate, where the minimum/maximum learning rate and the maximum adjusted iteration are set to $10^{-5}/10^{-4}$ and 20, respectively. The batch size is set to 12, and the maximum training epoch is 100. During training, we resize each image to 352×352 and feed it into DGNet with four data augmentation techniques: color enhancement, random flipping, random cropping, and random rotation. Finally, our DGNet and DGNet-S take 8.8 and 7.9 hours to reach the network convergence.

Table 2 Hyper-parameter settings of the proposed DGNet-S and DGNet

| Model | Backbone | $C_i$ | $C_g$ | $M$ | $\{N_1, N_2, N_3\}$ |
|---|---|---|---|---|---|
| DGNet-S | EfficientNet-B1 | 32 | 32 | 8 | $\{8, 16, 32\}$ |
| DGNet | EfficientNet-B4 | 64 | 32 | 8 | $\{4, 8, 16\}$ |

**Testing settings.** Once the network is well-trained, we resize the input images to 352×352 and test our DGNet-S and DGNet on three unseen test datasets. We take the final output $\boldsymbol{P}^C$ as the prediction map without any heuristic post-processing techniques, such as DenseCRF[55].

## 4 Experiments

### 4.1 Benchmarking

**Datasets.** There are three popular datasets in the COD field: 1) CAMO[33] has 1 250 camouflaged images and is divided into CAMO-Tr (1 000 samples) and CAMO-Te (250 samples). 2) COD10K[2] is the largest COD dataset so far, consisting of COD10K-Tr (3 040 images) and COD10K-Te (2 026 images). It is downloaded from multiple free photography websites, covering five super-classes and 69 sub-classes. 3) NC4K-Te[15], the largest test dataset, includes 4 121 samples, which are used to evaluate the models' generalization ability. Following the protocol of [2], we train our model on the hybrid dataset (i.e., COD10K-Tr + CAMO-Tr) with 4 040 samples and evaluate our method on the above three benchmarks (see Table 3).

**Metrics.** Following [2], we use five commonly used metrics for the evaluation: structure measure ($\mathcal{S}_\alpha$)[64], enhanced-alignment measure ($E_\phi$)[65, 66], F-measure ($F_\beta$)[67, 68], weighted F-measure ($F_\beta^w$)[69], and mean absolute error ($\mathcal{M}$). Besides, the precision-recall (PR) curves[67] are obtained by varying different thresholds from $[0, 255]$. Similar to this thresholding strategy, F-measure and E-measure curves are also reported. Moreover, we adopt three criteria to measure the model's complexity[2] and efficiency: the number of model parameters, measured in Millions (M), the number of multiply-accumulate (MACs) operations, measured in Giga (G), and inference speed measured in frames per second (fps).

**Competitors.** We compare our model with 20 SOTA

[2] The model's parameter and MACs are measured by the toolbox: https://github.com/sovrasov/flops-counter.pytorch.

Table 3 Quantitative results in terms of full metrics for cutting-edge competitors, including 8 SOD-related and 12 COD-related, on three COD-related test datasets. @$\mathcal{R}$ is the ranking of the current metric, and Mean@$\mathcal{R}$ indicates the mean ranking of all metrics. Note that $E_\phi^{mx}/F_\beta^{mx}$, $E_\phi^{mn}/F_\beta^{mn}$, and $E_\phi^{ad}/F_\beta^{ad}$ denote the maximum, mean, and adaptive value of E-measure/F-measure, respectively. ↑/↓ denotes that the higher/lower the score, the better. All the benchmark results are available at OneDrive-3.54 GB (https://anu365-my.sharepoint.com/:u:/g/personal/u7248002_anu_edu_au/EXLiBgp9nGNApBw9im2xznsBJ_ryGEW7hkJlL92gNaRAAg?e=RnmuiN).

| | | SOD-related models | | | | | | | | | | | COD-related models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EGNet | SCRN | CPD | CSNet-R | F3Net | UCNet | ITSD | MINet | SINet | PraNet | BAS | C2FNet | TINet | UGTR | PFNet | S-MGL | R-MGL | LSR | JCSOD | SINetV2 | **DGNet-S** | **DGNet** |
| | Metric | [56] | [57] | [58] | [59] | [48] | [60] | [61] | [62] | [1] | [3] | [63] | [28] | [37] | [21] | [14] | [18] | [18] | [15] | [22] | [2] | **Ours** | |
| NC4K-Te[15] | $\mathcal{S}_\alpha \uparrow$ | 0.777 | 0.830 | 0.788 | 0.750 | 0.780 | 0.811 | 0.811 | 0.805 | 0.808 | 0.822 | 0.817 | 0.838 | 0.829 | 0.839 | 0.829 | .829 | 0.833 | .840 | 0.842 | 0.847 | 0.845 | 0.857 |
| | @$\mathcal{R}$ | 21 | 9 | 19 | 22 | 20 | 16 | 15 | 18 | 17 | 13 | 14 | 7 | 12 | 6 | 11 | 10 | 8 | 5 | 4 | 2 | 3 | 1 |
| | $E_\phi^{mx} \uparrow$ | 0.864 | 0.897 | 0.865 | 0.793 | 0.848 | 0.886 | 0.883 | 0.881 | .883 | 0.888 | 0.872 | 0.904 | 0.890 | 0.899 | 0.898 | 0.893 | .893 | 0.907 | 0.907 | 0.914 | 0.913 | 0.922 |
| | @$\mathcal{R}$ | 20 | 9 | 19 | 22 | 21 | 14 | 15 | 17 | 16 | 13 | 18 | 6 | 12 | 7 | 8 | 11 | 10 | 5 | 4 | 2 | 3 | 1 |
| | $E_\phi^{mn} \uparrow$ | 0.841 | 0.854 | 0.804 | 0.773 | 0.824 | 0.871 | 0.845 | 0.846 | 0.871 | 0.876 | 0.859 | 0.897 | 0.879 | 0.874 | 0.888 | 0.863 | 0.867 | 0.895 | 0.898 | 0.903 | 0.902 | 0.911 |
| | @$\mathcal{R}$ | 19 | 16 | 21 | 22 | 20 | 11 | 18 | 17 | 12 | 9 | 15 | 5 | 8 | 10 | 7 | 14 | 13 | 6 | 4 | 2 | 3 | 1 |
| | $E_\phi^{ad} \uparrow$ | 0.826 | 0.864 | 0.842 | 0.812 | 0.853 | 0.883 | 0.855 | 0.876 | 0.882 | 0.871 | 0.868 | 0.898 | 0.880 | 0.886 | 0.892 | 0.884 | 0.889 | 0.902 | 0.903 | 0.898 | 0.899 | 0.907 |
| | @$\mathcal{R}$ | 21 | 17 | 20 | 22 | 19 | 11 | 18 | 14 | 12 | 15 | 16 | 6 | 13 | 9 | 7 | 10 | 8 | 3 | 2 | 5 | 4 | 1 |
| | $F_\beta^w \uparrow$ | 0.639 | 0.698 | 0.632 | 0.603 | 0.656 | 0.729 | 0.680 | 0.705 | 0.723 | 0.724 | 0.732 | 0.762 | 0.734 | 0.747 | 0.745 | 0.731 | .740 | 0.766 | 0.771 | 0.770 | 0.764 | 0.784 |
| | @$\mathcal{R}$ | 20 | 17 | 21 | 22 | 19 | 13 | 18 | 16 | 15 | 14 | 11 | 6 | 10 | 7 | 8 | 12 | 9 | 4 | 2 | 3 | 5 | 1 |
| | $F_\beta^{mx} \uparrow$ | 0.731 | 0.793 | 0.738 | 0.669 | 0.719 | 0.782 | 0.762 | 0.768 | 0.775 | 0.786 | 0.782 | 0.810 | 0.793 | 0.807 | 0.799 | 0.797 | 0.800 | 0.815 | 0.816 | 0.823 | 0.819 | 0.833 |
| | @$\mathcal{R}$ | 20 | 11 | 19 | 22 | 21 | 15 | 18 | 17 | 16 | 13 | 14 | 6 | 12 | 7 | 9 | 10 | 8 | 5 | 4 | 2 | 3 | 1 |
| | $F_\beta^{mn} \uparrow$ | 0.696 | 0.757 | 0.695 | 0.655 | 0.705 | 0.775 | 0.729 | 0.753 | 0.769 | 0.762 | 0.772 | 0.795 | 0.773 | 0.787 | 0.784 | 0.777 | 0.782 | 0.804 | 0.806 | 0.805 | 0.799 | 0.814 |
| | @$\mathcal{R}$ | 20 | 16 | 21 | 22 | 19 | 11 | 18 | 17 | 14 | 15 | 13 | 6 | 12 | 7 | 8 | 10 | 9 | 4 | 2 | 3 | 5 | 1 |
| | $F_\beta^{ad} \uparrow$ | 0.671 | 0.744 | 0.709 | 0.672 | 0.710 | 0.776 | 0.717 | 0.763 | 0.768 | 0.753 | 0.767 | 0.788 | 0.766 | 0.779 | 0.779 | 0.771 | 0.778 | 0.802 | 0.803 | 0.792 | 0.789 | 0.803 |
| | @$\mathcal{R}$ | 22 | 17 | 20 | 21 | 19 | 10 | 18 | 15 | 12 | 16 | 13 | 6 | 14 | 8 | 7 | 11 | 9 | 3 | 2 | 4 | 5 | 1 |
| | $\mathcal{M} \downarrow$ | .075 | 0.059 | 0.074 | 0.088 | 0.070 | 0.055 | 0.064 | 0.060 | .058 | 0.059 | 0.058 | 0.049 | 0.055 | 0.052 | 0.053 | 0.055 | 0.052 | 0.048 | 0.047 | 0.048 | 0.047 | 0.042 |
| | @$\mathcal{R}$ | 21 | 15 | 20 | 22 | 19 | 10 | 18 | 17 | 14 | 16 | 13 | 6 | 12 | 8 | 9 | 11 | 7 | 5 | 2 | 4 | 3 | 1 |
| | Mean@$\mathcal{R}$ | 21 | 15 | 20 | 22 | 19 | 12 | 18 | 17 | 16 | 13 | 14 | 6 | 11 | 7 | 8 | 10 | 9 | 5 | 2 | 3 | 4 | 1 |
| CAMO-Te[33] | $\mathcal{S}_\alpha \uparrow$ | 0.732 | 0.779 | 0.726 | 0.771 | 0.711 | 0.739 | 0.750 | 0.737 | 0.745 | 0.769 | 0.749 | 0.796 | 0.781 | 0.785 | 0.782 | 0.772 | 0.775 | 0.787 | 0.800 | 0.820 | 0.826 | 0.839 |
| | @$\mathcal{R}$ | 20 | 10 | 21 | 13 | 22 | 18 | 15 | 19 | 17 | 14 | 16 | 5 | 9 | 7 | 8 | 12 | 11 | 6 | 4 | 3 | 2 | 1 |
| | $E_\phi^{mx} \uparrow$ | 0.820 | 0.850 | 0.801 | 0.849 | 0.780 | 0.820 | 0.830 | .818 | 0.829 | 0.837 | 0.808 | 0.864 | 0.848 | 0.854 | 0.855 | 0.842 | 0.842 | 0.854 | .873 | 0.895 | 0.907 | 0.915 |
| | @$\mathcal{R}$ | 17 | 9 | 21 | 10 | 22 | 18 | 15 | 19 | 16 | 14 | 20 | 5 | 11 | 8 | 6 | 13 | 12 | 7 | 4 | 3 | 2 | 1 |
| | $E_\phi^{mn} \uparrow$ | 0.800 | 0.797 | 0.723 | 0.795 | 0.741 | 0.787 | 0.780 | 0.767 | 0.804 | 0.824 | 0.796 | 0.854 | 0.836 | 0.823 | 0.842 | 0.807 | 0.812 | 0.838 | 0.859 | 0.882 | 0.893 | 0.901 |
| | @$\mathcal{R}$ | 14 | 15 | 22 | 17 | 21 | 18 | 19 | 20 | 13 | 9 | 16 | 5 | 8 | 10 | 6 | 12 | 11 | 7 | 4 | 3 | 2 | 1 |
| | $E_\phi^{ad} \uparrow$ | 0.811 | 0.848 | 0.810 | .847 | 0.802 | 0.811 | 0.830 | 0.826 | .825 | 0.833 | 0.806 | 0.864 | 0.845 | 0.859 | 0.852 | 0.850 | 0.847 | 0.855 | 0.865 | 0.875 | 0.892 | 0.901 |
| | @$\mathcal{R}$ | 18 | 10 | 20 | 12 | 22 | 19 | 15 | 16 | 17 | 14 | 21 | 5 | 13 | 6 | 8 | 9 | 11 | 7 | 4 | 3 | 2 | 1 |
| | $F_\beta^w \uparrow$ | 0.604 | 0.643 | 0.553 | 0.642 | 0.564 | 0.640 | 0.610 | 0.613 | 0.644 | 0.663 | 0.646 | 0.719 | 0.678 | 0.686 | 0.695 | 0.664 | 0.673 | 0.696 | 0.728 | 0.743 | 0.754 | 0.769 |
| | @$\mathcal{R}$ | 20 | 15 | 22 | 16 | 21 | 17 | 19 | 18 | 14 | 12 | 13 | 5 | 9 | 8 | 7 | 11 | 10 | 6 | 4 | 3 | 2 | 1 |
| | $F_\beta^{mx} \uparrow$ | 0.688 | 0.738 | 0.667 | 0.740 | 0.630 | 0.708 | 0.694 | 0.683 | 0.708 | 0.728 | 0.703 | 0.771 | 0.745 | 0.754 | 0.758 | 0.739 | 0.740 | 0.753 | 0.779 | 0.801 | 0.810 | 0.822 |
| | @$\mathcal{R}$ | 19 | 13 | 21 | 10 | 22 | 15 | 18 | 20 | 16 | 14 | 17 | 5 | 9 | 7 | 6 | 12 | 11 | 8 | 4 | 3 | 2 | 1 |

Table 3 (continued) Quantitative results in terms of full metrics for cutting-edge competitors, including 8 SOD-related and 12 COD-related, on three COD-related test datasets. @$\mathcal{R}$ is the ranking of the current metric, and Mean@$\mathcal{R}$ indicates the mean ranking of all metrics. Note that $E_\phi^{mx}/F_\beta^{mx}$, $E_\phi^{mn}/F_\beta^{mn}$, and $E_\phi^{ad}/F_\beta^{ad}$ denote the maximum, mean, and adaptive value of E-measure/F-measure, respectively. ↑/↓ denotes that the higher/lower the score, the better. All the benchmark results are available at OneDrive-3.54 GB (https://anu365-my.sharepoint.com/:u:/g/personal/u7248002_anu_edu_au/EXLiBgp9nGNApBw9im2xznsBJ_ryGEW7hkJlL92gNaRAAg?e=RnmuiN).

| | | SOD-related models | | | | | | | | | | | COD-related models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EGNet | SCRN | CPD | CSNet-R | F3Net | UCNet | ITSD | MINet | SINet | PraNet | BAS | C2FNet | TINet | UGTR | PFNet | S-MGL | R-MGL | LSR | JCSOD | SINetV2 | **DGNet-S** | **DGNet** |
| | Metric | [56] | [57] | [58] | [59] | [48] | [60] | [61] | [62] | [1] | [3] | [63] | [28] | [37] | [21] | [14] | [18] | [18] | [15] | [22] | [2] | **Ours** | |
| COD10K-Te[2] | $F_\beta^{mn}$ ↑ | 0.670 | 0.705 | 0.614 | 0.705 | 0.616 | 0.700 | 0.663 | 0.667 | 0.702 | 0.710 | 0.692 | 0.762 | 0.728 | 0.738 | 0.746 | 0.721 | 0.726 | 0.744 | 0.772 | 0.782 | 0.792 | 0.806 |
| | @$\mathcal{R}$ | 18 | 13 | 22 | 14 | 21 | 16 | 20 | 19 | 15 | 12 | 17 | 5 | 9 | 8 | 6 | 11 | 10 | 7 | 4 | 3 | 2 | 1 |
| | $F_\beta^{ad}$ ↑ | 0.667 | 0.733 | 0.678 | 0.730 | 0.661 | 0.716 | 0.692 | 0.704 | 0.712 | 0.715 | 0.696 | 0.764 | 0.729 | 0.749 | 0.751 | 0.733 | 0.738 | 0.756 | 0.779 | 0.779 | 0.786 | 0.804 |
| | @$\mathcal{R}$ | 21 | 10 | 20 | 12 | 22 | 14 | 19 | 17 | 16 | 15 | 18 | 5 | 13 | 8 | 7 | 11 | 9 | 6 | 4 | 3 | 2 | 1 |
| | $\mathcal{M}$ ↓ | 0.109 | 0.090 | 0.114 | 0.092 | 0.109 | 0.094 | 0.102 | 0.096 | 0.092 | 0.094 | .096 | 0.080 | 0.087 | 0.086 | 0.085 | 0.089 | 0.088 | 0.080 | 0.073 | 0.070 | 0.063 | 0.057 |
| | @$\mathcal{R}$ | 21 | 12 | 22 | 13 | 20 | 15 | 19 | 18 | 14 | 16 | 17 | 6 | 9 | 8 | 7 | 11 | 10 | 5 | 4 | 3 | 2 | 1 |
| | Mean@$\mathcal{R}$ | 20 | 12 | 21 | 13 | 22 | 16 | 18 | 19 | 15 | 14 | 17 | 5 | 9 | 8 | 7 | 11 | 10 | 6 | 4 | 3 | 2 | 1 |
| | $\mathcal{S}_\alpha$ ↑ | 0.736 | 0.789 | 0.748 | 0.778 | .739 | 0.776 | 0.767 | 0.769 | 0.776 | 0.789 | 0.802 | 0.813 | 0.793 | 0.818 | 0.800 | 0.811 | 0.814 | 0.804 | 0.809 | 0.815 | 0.810 | 0.822 |
| | @$\mathcal{R}$ | 22 | 13 | 20 | 15 | 21 | 17 | 19 | 18 | 16 | 14 | 10 | 5 | 12 | 2 | 11 | 6 | 4 | 9 | 8 | 3 | 7 | 1 |
| | $E_\phi^{mx}$ ↑ | .855 | 0.880 | 0.842 | 0.871 | 0.819 | 0.867 | 0.861 | 0.864 | 0.874 | 0.879 | 0.870 | 0.900 | 0.878 | 0.891 | 0.890 | 0.890 | 0.890 | 0.892 | 0.891 | 0.906 | 0.905 | 0.911 |
| | @$\mathcal{R}$ | 20 | 11 | 21 | 15 | 22 | 17 | 19 | 18 | 14 | 12 | 16 | 4 | 13 | 7 | 10 | 9 | 8 | 5 | 6 | 2 | 3 | 1 |
| | $E_\phi^{mn}$ ↑ | 0.810 | 0.817 | 0.766 | 0.810 | 0.795 | 0.857 | 0.808 | 0.823 | 0.864 | 0.861 | 0.855 | 0.890 | 0.861 | 0.853 | 0.877 | 0.845 | 0.852 | 0.880 | 0.884 | 0.887 | 0.888 | 0.896 |
| | @$\mathcal{R}$ | 19 | 17 | 22 | 18 | 21 | 11 | 20 | 16 | 8 | 10 | 12 | 2 | 9 | 13 | 7 | 15 | 14 | 6 | 5 | 4 | 3 | 1 |
| | $E_\phi^{ad}$ ↑ | 0.753 | 0.789 | 0.768 | 0.791 | 0.818 | 0.867 | 0.787 | 0.837 | 0.867 | 0.839 | 0.869 | 0.886 | 0.848 | 0.850 | 0.868 | 0.851 | 0.865 | 0.882 | 0.882 | 0.863 | 0.868 | 0.877 |
| | @$\mathcal{R}$ | 22 | 19 | 21 | 18 | 17 | 8 | 20 | 16 | 9 | 15 | 5 | 1 | 14 | 13 | 7 | 12 | 10 | 3 | 2 | 11 | 6 | 4 |
| | $F_\beta^{w}$ ↑ | 0.517 | 0.575 | 0.509 | 0.569 | 0.544 | 0.633 | 0.557 | 0.601 | 0.631 | 0.629 | 0.677 | 0.686 | 0.635 | 0.667 | 0.660 | 0.655 | 0.666 | 0.673 | 0.684 | 0.680 | 0.672 | 0.693 |
| | @$\mathcal{R}$ | 21 | 17 | 22 | 18 | 20 | 13 | 19 | 16 | 14 | 15 | 5 | 2 | 12 | 8 | 10 | 11 | 9 | 6 | 3 | 4 | 7 | 1 |
| | $F_\beta^{mx}$ ↑ | 0.633 | 0.699 | 0.634 | 0.679 | 0.609 | 0.691 | 0.658 | 0.672 | 0.691 | 0.704 | 0.729 | 0.743 | 0.712 | 0.742 | 0.725 | 0.733 | 0.738 | 0.732 | 0.738 | 0.752 | 0.743 | 0.759 |
| | @$\mathcal{R}$ | 21 | 14 | 20 | 17 | 22 | 16 | 19 | 18 | 15 | 13 | 10 | 3 | 12 | 5 | 11 | 8 | 7 | 9 | 6 | 2 | 4 | 1 |
| | $F_\beta^{mn}$ ↑ | 0.582 | 0.651 | 0.582 | 0.635 | 0.593 | 0.681 | 0.615 | 0.654 | 0.679 | 0.671 | 0.715 | 0.723 | 0.679 | 0.712 | 0.701 | 0.702 | 0.711 | 0.715 | 0.721 | 0.718 | 0.710 | 0.728 |
| | @$\mathcal{R}$ | 22 | 17 | 21 | 18 | 20 | 12 | 19 | 16 | 13 | 15 | 6 | 2 | 14 | 7 | 11 | 10 | 8 | 5 | 3 | 4 | 9 | 1 |
| | $F_\beta^{ad}$ ↑ | 0.526 | 0.593 | 0.555 | 0.589 | 0.588 | 0.673 | 0.573 | 0.639 | 0.667 | 0.640 | 0.707 | 0.703 | 0.652 | 0.671 | 0.676 | 0.667 | 0.681 | 0.699 | 0.705 | 0.682 | 0.680 | 0.698 |
| | @$\mathcal{R}$ | 22 | 17 | 21 | 18 | 19 | 10 | 20 | 16 | 12 | 15 | 1 | 3 | 14 | 11 | 9 | 13 | 7 | 4 | 2 | 6 | 8 | 5 |
| | $\mathcal{M}$ ↓ | 0.061 | 0.047 | 0.058 | 0.047 | 0.051 | 0.042 | 0.051 | 0.043 | 0.043 | 0.045 | 0.038 | 0.036 | 0.042 | 0.035 | 0.040 | 0.037 | 0.035 | 0.037 | .035 | 0.037 | 0.036 | 0.033 |
| | @$\mathcal{R}$ | 22 | 18 | 21 | 17 | 20 | 13 | 19 | 15 | 14 | 16 | 10 | 5 | 12 | 4 | 11 | 9 | 3 | 8 | 2 | 7 | 6 | 1 |
| | Mean@$\mathcal{R}$ | 22 | 16 | 21 | 18 | 20 | 14 | 19 | 17 | 13 | 15 | 9 | 2 | 12 | 8 | 10 | 11 | 7 | 6 | 3 | 4 | 5 | 1 |

competitors (see Table 3), including 8 SOD-based and 12 COD-based. For a fair comparison, all results were either taken from the public website or produced by retraining the models on the same training dataset with default settings.

## 4.2 Results and analysis

**Quantitative results.** As shown in Table 3, DGNet achieves promising performance in terms of all metrics. Especially, the gradient-based learning strategy helps to improve the completeness of the predictions, providing a 2.6% gain of $F_\beta^w$ on CAMO-Te than rank@1 model SINetV2[2].

**Quantitative curves.** As shown in Fig. 5, we plot the precision-recall (1st row), F-measure (2nd row), and E-measure (3rd row) curves of all COD-related competitors by varying with different thresholds. All comparisons show that our curves with magenta solid/dotted lines are significantly better than other methods on three datasets.

**Qualitative results.** The visual comparisons of four top-tier COD baselines and our DGNet are shown in

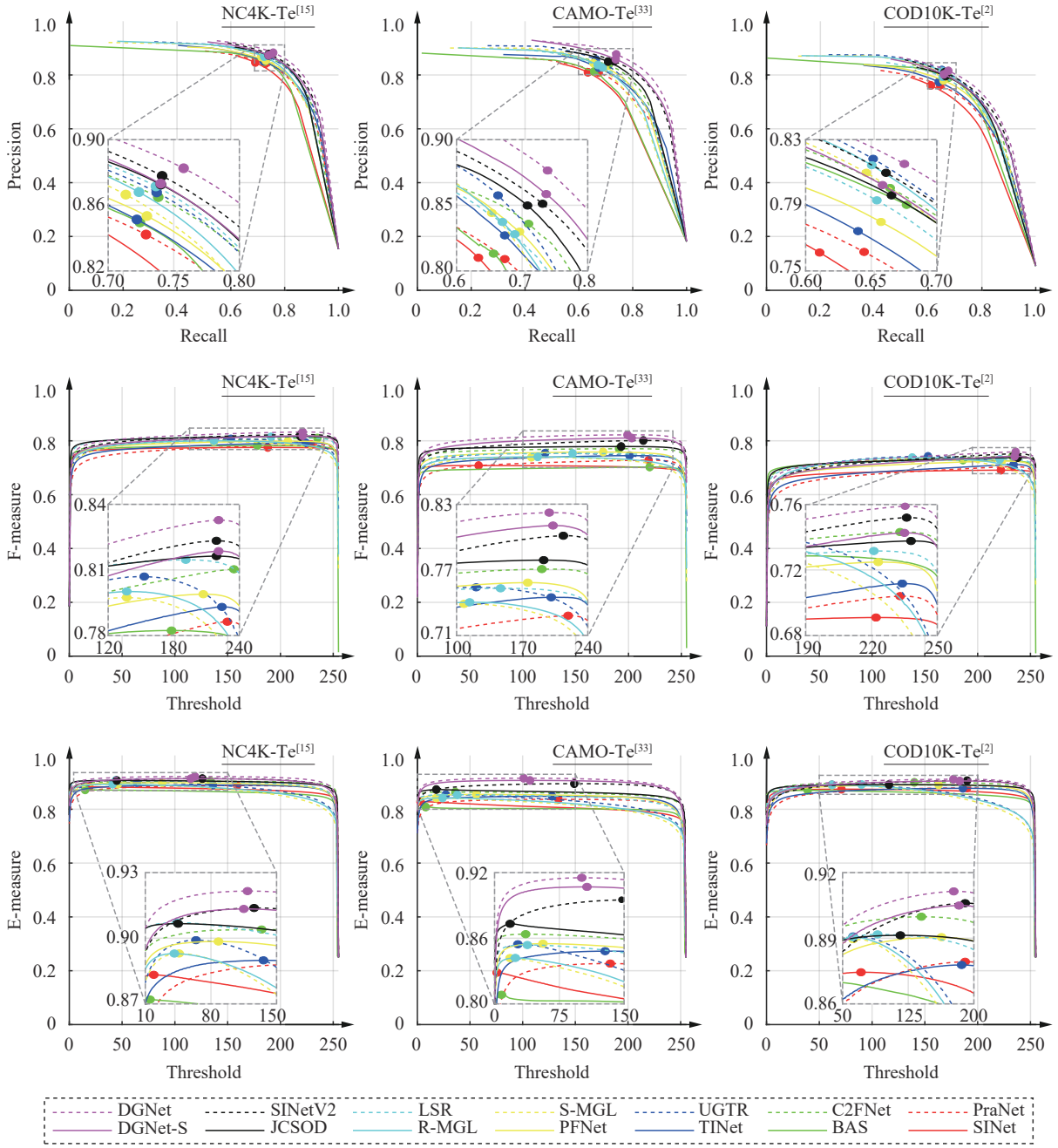



Fig. 5 PR curves (1st row), F-measure curves (2nd row), and E-measure curves (3rd row) of COD-related competitors on three popular datasets. The closer the PR curve is to the upper-right corner, the better the performance is. The higher the F-measure/E-measure curve is, the better the performance and the better the model works. Best viewed in color.

Fig. 6. Interestingly, these competitors fail to provide complete segmentation results for the camouflaged objects touching the image boundary. By contrast, our approach can precisely locate the target region and provide exact predictions due to the gradient learning strategy.

**Efficiency analysis.** To better unveil the trade-off, two instances consistently obtain the best trade-off compared to existing competitors (see Fig. 7). DGNet outperforms the cutting-edge model SINetV2[2] with a large margin ($F_\beta^w$: +2.6%). Notably, our efficient instance DGNet-S performs better than JCSOD[22], with 113.33 M fewer parameters. Besides, we also report the runtime comparison of all COD-related competitors in Table 4, which are tested on an NVIDIA RTX TITAN GPU. It clearly illustrates that DGNet-S and DGNet can achieve super real-time inference speeds (i.e., 80 fps & 58 fps).

## 4.3 Ablation study

We further ablate the core modules to verify the effectiveness of each part and configuration. For ecological reasons, we select the DGNet-S as the base model in this section.

**Contribution of base network.** In Table 5(a), we remove the texture encoder and GIT from DGNet-S and term it as the base network (#01). Compared to it, our DGNet-S (#**S**) significantly improves the performance while slightly increasing the model parameters by 0.06 M.

**Configuration of dimensional reduction.** We change the channel $C_i$ to 16 (#02), 32 (#**S**), 64 (#03),

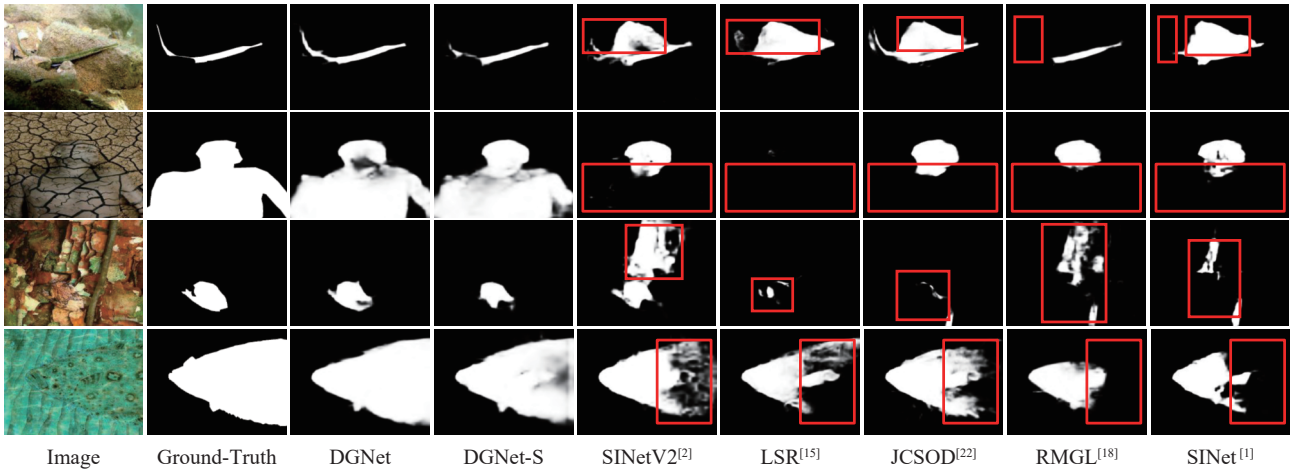



Fig. 6 Visualization of popular COD baselines and the proposed DGNet. Red boxes denote false-positive/false-negative predictions. More results are presented at https://github.com/GewelsJI/DGNet.

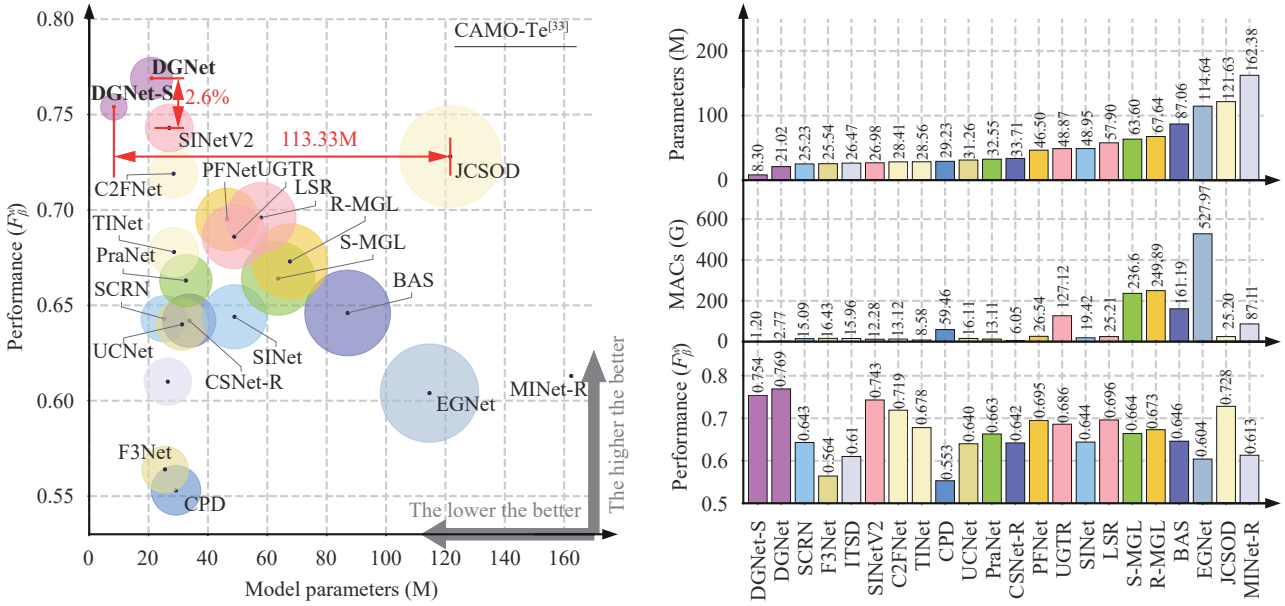



Fig. 7 (Left) We present the scatter relationship between the performance ($F_\beta^w$) and the parameters of all competitors on CAMO-Te[33]. The larger the colored scatter point size, the heavier the model parameters. (Right) We also report the parallel histogram comparison of the model's parameters, MACs, and performance ($F_\beta^w$). Best viewed in color.

Table 4 Inference speed (fps) among 12 COD-related models and the proposed two instances (i.e., DGNet-S and DGNet)

| Model | **DGNet-S** | **DGNet** | SINetV2[2] | JCSOD[22] | LSR[15] | R-MGL[18] | S-MGL[18] |
|---|---|---|---|---|---|---|---|
| Input size | 352×352 | 352×352 | 352 × 352 | 352×352 | 352×352 | 473×473 | 473×473 |
| Speed (fps) | **80** | 58 | 68 | 43 | 31 | 9 | 13 |
| Model | PFNet[14] | UGTR[21] | TINet[37] | C2FNet[28] | BAS[63] | PraNet[3] | SINet[1] |
| Input size | 416×416 | 473×473 | 352×352 | 352×352 | 288×288 | 352×352 | 352×352 |
| Speed (fps) | 78 | 15 | 50 | 68 | 31 | 73 | 63 |

Table 5 Ablation studies. #Para and #MACs denote the parameters and multiply-accumulate operations of the model.

| | | Efficiency | | NC4K-Te[15] | | | CAMO-Te[33] | | | COD10K-Te[2] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. | Variant | #Para | #MACs | $\mathcal{S}_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $\mathcal{M} \downarrow$ | $\mathcal{S}_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $\mathcal{M} \downarrow$ | $\mathcal{S}_\alpha \uparrow$ | $F_\beta^w \uparrow$ | $\mathcal{M} \downarrow$ |
| #**S** | **DGNet-S** | 8.30M | 1.20G | 0.845 | 0.764 | 0.047 | 0.826 | 0.754 | 0.063 | 0.810 | 0.672 | 0.036 |
| (a) Base network → see Section 3.1 | | | | | | | | | | | | |
| #01 | Base | 8.24M | 0.58G | 0.834 | 0.676 | 0.061 | 0.814 | 0.670 | 0.072 | 0.793 | 0.550 | 0.049 |
| (b) Configuration of dimensional reduction → see Section 3.1 | | | | | | | | | | | | |
| #02 | $C_i = 16$ | 8.00M | 0.81G | 0.842 | 0.758 | 0.048 | 0.824 | 0.749 | 0.066 | 0.806 | 0.663 | 0.037 |
| #03 | $C_i = 64$ | 9.36M | 2.69G | 0.845 | 0.764 | 0.047 | 0.827 | 0.748 | 0.065 | 0.812 | 0.673 | 0.036 |
| #04 | $C_i = 128$ | 13.30M | 8.55G | 0.847 | 0.768 | 0.046 | 0.828 | 0.751 | 0.062 | 0.810 | 0.672 | 0.036 |
| (c) Network decoupling strategy → see Section 3.2 | | | | | | | | | | | | |
| #05 | $w/\ \mathbf{X}_2$ | 8.24M | 0.59G | 0.840 | 0.712 | 0.055 | 0.822 | 0.701 | 0.074 | 0.805 | 0.597 | 0.043 |
| (d) Should we use $\mathbf{Z}^G$ as supervision? → see Equ. (1) | | | | | | | | | | | | |
| #06 | $w/\ \mathbf{Z}^B$ | 8.30M | 1.20G | 0.841 | 0.753 | 0.049 | 0.821 | 0.737 | 0.067 | 0.804 | 0.654 | 0.038 |
| (e) Group number $M$ → see Equ. (2) | | | | | | | | | | | | |
| #07 | $M = 1$ | 8.30M | 1.20G | 0.841 | 0.756 | 0.049 | 0.822 | 0.751 | 0.064 | 0.806 | 0.662 | 0.037 |
| #08 | $M = 4$ | 8.30M | 1.20G | 0.842 | 0.759 | 0.048 | 0.822 | 0.742 | 0.067 | 0.809 | 0.669 | 0.036 |
| #09 | $M = 16$ | 8.30M | 1.20G | 0.842 | 0.752 | 0.049 | 0.829 | 0.744 | 0.065 | 0.803 | 0.651 | 0.039 |
| #10 | $M = 32$ | 8.30M | 1.20G | 0.845 | 0.913 | 0.047 | 0.827 | 0.745 | 0.063 | 0.809 | 0.666 | 0.036 |
| (f) Scaling factors $N \in \{N_1, N_2, N_3\}$ → see Equ. (5) | | | | | | | | | | | | |
| #11 | $\{2, 4, 8\}$ | 8.31M | 1.20G | 0.842 | 0.755 | 0.048 | 0.821 | 0.741 | 0.065 | 0.808 | 0.663 | 0.036 |
| #12 | $\{4, 8, 16\}$ | 8.30M | 1.20G | 0.844 | 0.762 | 0.047 | 0.823 | 0.744 | 0.065 | 0.806 | 0.666 | 0.037 |
| (g) More sub-branches in soft grouping strategy → see Equ. (5) | | | | | | | | | | | | |
| #13 | $N \in \{4, 8, 16, 32\}$ | 8.30M | 1.20G | 0.844 | 0.760 | 0.048 | 0.829 | 0.748 | 0.064 | 0.811 | 0.669 | 0.037 |
| #14 | $N \in \{2, 4, 8, 16, 32\}$ | 8.31M | 1.20G | 0.846 | 0.765 | 0.047 | 0.825 | 0.750 | 0.063 | 0.810 | 0.670 | 0.037 |
| (h) Gradient-induced transition → see Equ. (6) | | | | | | | | | | | | |
| #15 | $w/o\ \mathcal{T}_i$ | 8.31M | 1.20G | 0.839 | 0.748 | 0.050 | 0.825 | 0.741 | 0.065 | 0.802 | 0.649 | 0.039 |

and 128 (#04) and find that more parameters may lead to performance saturation. To achieve the best trade-off between resource and speed, we choose $C_i$=32 as the default setting.

**Contribution of network decoupling strategy.** We explore the necessity of our decoupling strategy. Inspired by [19], we replace the feature extracted from the texture encoder with the low-level feature $\boldsymbol{X}_2$ from the context encoder, which yields a single-stream network (#05). Notably, we only change the extraction manner of texture features and preserve the gradient-wise supervision for both variants (i.e., #05 & #**S**) to ensure unbiased ablation. Table 5(c) demonstrates that decoupling the network into two streams can improve the performance ($F_\beta^w$: +5.3% on CAMO-Te), which benefits from the modelling of separated branches without feature ambiguity in different hierarchies.

**Contribution of object gradient supervision.** We

replace the gradient map $\boldsymbol{Z}^G$ (#**S**) with the boundary mask $\boldsymbol{Z}^B$ (#06) to supervise the context learning process. The improvement ($F_\beta^w$: +1.7% on CAMO-Te) of our gradient map supervision further demonstrates the effectiveness. The first row of Fig. 8 presents the low-level features extracted from the texture learning branch under different supervision types. It shows that our solution can enforce the network to capture the gradient-sensitive information inside the camouflaged object's body, where those pixels learn to draw the observer's attention.

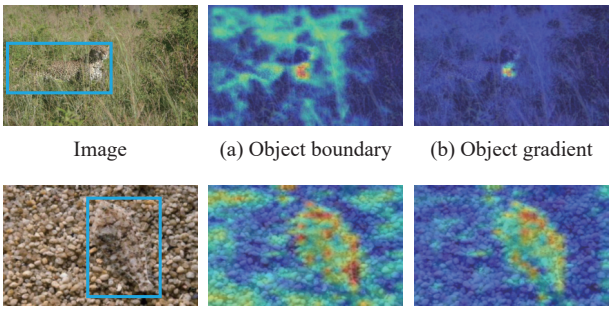



Fig. 8 Feature visualizations of two core designs, including the supervision of object gradient (1st row) and the GIT (2nd row).

We further experiment using the supervision of texture labels[37] (see Fig. 2). The results in Table 6 demonstrate that our gradient-supervision manner (i.e., w/ DGNet-grad) is better than the texture-supervision (i.e., w/ TINet-text). Besides, our method is simpler and more efficient than TINet, e.g., DGNet-S (8.0M) VS. TINet (28.6M), DGNet-S (80 fps) VS. TINet (50 fps). With such a compact design, we also achieve the new SOTA performance on CAMO-Te, e.g., DGNet-S ($\mathcal{S}_\alpha = 0.826$), DGNet ($\mathcal{S}_\alpha = 0.839$) VS. TINet ($\mathcal{S}_\alpha = 0.781$).

Table 6 Training DGNet-S under the supervision of texture label (TINet-text[37]) and our object gradient label (DGNet-grad)

| | NC4K-Te | | | CAMO-Te | | | COD10K-Te | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\mathcal{S}_\alpha$ | $F_\beta^w$ | $\mathcal{M}$ | $\mathcal{S}_\alpha$ | $F_\beta^w$ | $\mathcal{M}$ | $\mathcal{S}_\alpha$ | $F_\beta^w$ | $\mathcal{M}$ |
| w/TINet-text | 0.839 | 0.747 | 0.050 | 0.820 | 0.731 | 0.068 | 0.803 | 0.652 | 0.040 |
| **w/DGNet-grad** | **0.845** | **0.764** | **0.047** | **0.826** | **0.754** | **0.063** | **0.810** | **0.672** | **0.036** |

**Configuration of group numbers.** In Table 5(e), we report different variants with respect to four group numbers $M$, which are equal to 1 (#07), 4 (#08), 8 (#**S**), 16 (#09), and 32 (#10), respectively. Note that #07 ($M = 1$) means ungrouped candidate features, which causes degraded performance ($F_\beta^w$: −1.0% on COD10K-Te). We empirically choose $M = 8$ with the best performance.

**Configuration of scaling factors.** We also discuss how scaling factors $N \in \{N_1, N_2, N_3\}$ affect the model performance in Table 5(f). Compared with different configurations (#11: $\{2, 4, 8\}$ and #12: $\{4, 8, 16\}$), our finer-grained factors (#**S**: $\{8, 16, 32\}$) lead to better prediction performance. As shown in Fig. 4, we present the feature visualization of three parallel features (i.e., $\boldsymbol{A}_i^{N_1}$, $\boldsymbol{A}_i^{N_2}$, and $\boldsymbol{A}_i^{N_3}$), where the network puts different attention weights on different parts of the object inside. This also validates that parallel residual learning can enhance the context feature from different group-aware perspectives.

**Do we need more sub-branches for soft grouping?** As shown in Table 5(g), we set three ablative experiments for different sub-branches: three (#**S**: $N \in \{8, 16, 32\}$), four (#13: $N \in \{4, 8, 16, 32\}$), and five (#14: $N \in \{2, 4, 8, 16, 32\}$) sub-branches. The comparison results unveil that more sub-branches would present unstable performance on all the datasets.

**Contribution of gradient-induced transition.** We replace the whole GIT in our model with the naive channel-wise concatenation (#15: *w/o* $\mathcal{T}_i$ in Table 5(h)) to verify its effectiveness, which shows that our DGNet-S equipped with GIT (#**S**: *w/* $\mathcal{T}_i$) can improve 2.3% $F_\beta^w$ on the COD10K-Te dataset. Moreover, as shown in the second row of Fig. 8, the model obtains a cleaner and finer representation $\boldsymbol{Z}_i^{\mathrm{T}}$ (i.e., Fig. 8(d) after GIT) while suppressing the noises in the background of $\boldsymbol{X}_i^R$ (i.e., Fig. 8(c) before GIT). A clear benefit of the adaptive aggregation of the context and texture cues in the GIT.

## 4.4 Limitations

**Efficient backbone VS. lightweight.** We further validate the potential value of our method on limited hardware conditions by replacing the efficient backbone, EfficientNet[40], with a lightweight one, MobileNet[70]. The results, as in Table 7, show that our method achieves unsatisfactory performance with a lightweight backbone, i.e., MobNet-S (2.96 M) and MobNet-L (6.96 M), leaving a huge room for our future exploration.

Table 7 Our method with different backbones, including EfficientNet[40] (i.e., EffNet-B1 & EffNet-B4) VS. MobileNet[70] (i.e., MobNet-S & MobNet-L)

| | | | NC4K-Te | | CAMO-Te | | COD10K-Te | |
|---|---|---|---|---|---|---|---|---|
| | #Para | MACs | $\mathcal{S}_\alpha$ | $F_\beta^w$ | $\mathcal{S}_\alpha$ | $F_\beta^w$ | $\mathcal{S}_\alpha$ | $F_\beta^w$ |
| MobNet-S | 2.96M | 1.27G | 0.779 | 0.638 | 0.735 | 0.587 | 0.729 | 0.517 |
| **EffNet-B1** | 8.30M | 1.20G | 0.845 | 0.764 | 0.826 | 0.754 | 0.810 | 0.672 |
| MobNet-L | 6.96M | 3.17G | 0.820 | 0.723 | 0.791 | 0.686 | 0.780 | 0.620 |
| **EffNet-B4** | 21.02M | 2.77G | 0.857 | 0.784 | 0.839 | 0.769 | 0.822 | 0.693 |

**Challenging cases.** Despite our method's satisfactory performance, it may fail in challenging camouflaged scenarios as follows. First, we argue that in the proposed strategy, it is still difficult to provide enough texture cues in the limited small target region, resulting in false-positive predictions. As shown in Fig. 9, such cases also easily

confuse the rank@1 approach SINetV2[2], thus deserving further studies.

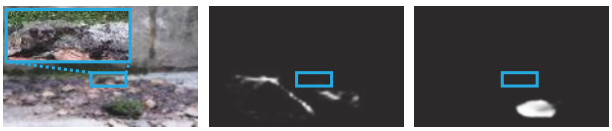



Fig. 9 Hard sample with a small camouflaged object

Second, we observe that not all the camouflaged objects with noticeable gradient changes inside themselves. As shown in the first row of Fig. 10, our method could segment a white rabbit with non-distinct gradient changes. However, our method fails under extreme conditions, as in the second row of Fig. 10, which has rare gradient cues. It needs to be designed by incorporating more heuristic and learnable patterns for future improvements.

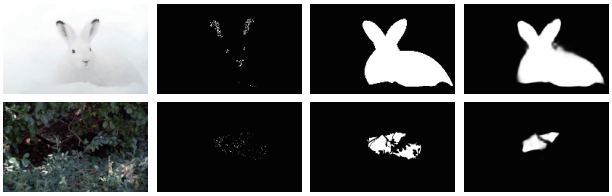



Fig. 10 Visual comparison of the object with non-distinct object gradient cues

Additionally, we noticed a recently released COD method, ZoomNet[34], after the submission. As shown in Table 8, our DGNet surpasses the ZoomNet by a margin (i.e., NC4K-Te: +1.3% $E_\phi^{mx}$ and CAMO-Te: +2.3% $E_\phi^{mx}$), but fails to outperform ZoomNet on COD10K-Te. ZoomNet occupies more computational costs (32.38 M parameters) than our DGNet (21.02 M parameters). It inspires us to incorporate the zooming strategy into our network for our future extension.

Table 8 Performance comparison of our DGNet and the recently released ZoomNet[34] on three testing datasets

| | | | NC4K-Te | | CAMO-Te | | COD10K-Te | |
|---|---|---|---|---|---|---|---|---|
| | #Para | MACs | $\mathcal{S}_\alpha$ | $E_\phi^{mx}$ | $\mathcal{S}_\alpha$ | $E_\phi^{mx}$ | $\mathcal{S}_\alpha$ | $E_\phi^{mx}$ |
| ZoomNet | 32.38 M | 34.96 G | 0.853 | 0.912 | 0.820 | 0.892 | **0.838** | **0.911** |
| **DGNet** | **21.02 M** | **2.77 G** | **0.857** | **0.922** | **0.839** | **0.915** | 0.822 | **0.911** |

## 5 Downstream applications

This section also assesses the generalization capabilities of three downstream applications.

**Polyp segmentation.** In the early diagnosis of colonoscopy, the low boundary contrast between a polyp and its highly-similar surroundings significantly decreases the detectability of colorectal cancer. To demonstrate the generality of our method in the medical field, we follow the same benchmark protocols as [3] and retrain our DGNet on the training set of Kvasir-SEG[71] and CVC-ClinicDB[72] datasets. We use two unseen test datasets: CVC-ColonDB[73] and ETIS-LPDB[74]. Table 9 shows that our †DGNet consistently surpasses four cutting-edge polyp segmentation methods in four metrics, including $\mathcal{S}_\alpha$, $E_\phi^{mx}$, $F_\beta^w$, and the maximum Dice score ($D^{mx}$). Notably, †DGNet denotes that we retrain DGNet on the task-specific training dataset. Fig. 11(a) shows the visualization results generated by our †DGNet.

Table 9 Quantitative results on two popular polyp segmentation test datasets

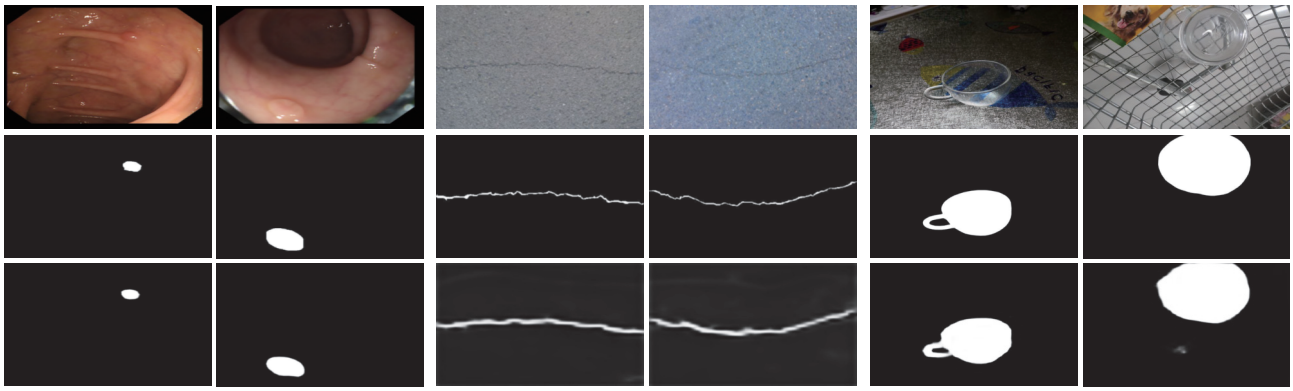
| | CVC-ColonDB[73] | | | | ETIS-LPDB[74] | | | |
|---|---|---|---|---|---|---|---|---|
| Baseline | $\mathcal{S}_\alpha \uparrow$ | $E_\phi^{mx} \uparrow$ | $F_\beta^w \uparrow$ | $D^{mx} \uparrow$ | $\mathcal{S}_\alpha \uparrow$ | $E_\phi^{mx} \uparrow$ | $F_\beta^w \uparrow$ | $D^{mx} \uparrow$ |
| UNet[75] | 0.710 | 0.781 | 0.491 | 0.560 | 0.684 | 0.740 | 0.366 | 0.444 |
| UNet++[76] | 0.692 | 0.764 | 0.467 | 0.550 | 0.683 | 0.776 | 0.390 | 0.509 |
| PraNet[3] | 0.820 | 0.872 | 0.699 | 0.728 | 0.794 | 0.841 | 0.600 | 0.639 |
| MSNet[77] | 0.838 | 0.883 | 0.736 | 0.766 | 0.845 | 0.890 | 0.677 | 0.736 |
| **†DGNet** | **0.858** | **0.898** | **0.765** | **0.789** | **0.847** | **0.904** | **0.690** | **0.741** |

**Defect detection.** Substandard products (e.g., tiles, wood) will inevitably incur unrecoverable economic losses in manufacturing. We further retrain our †DGNet on the road crack detection dataset (i.e., CrackForest[78]), using 60% of the samples for training and 40% for testing. Fig. 11(b) presents some visualization cases.

**Transparent object segmentation.** In daily life, intelligent agents such as robots and drones need to identify unnoticeable transparent objects (e.g., glasses, bottles, and mirrors) to avoid accidents. We also verify the effectiveness of the retrained model †DGNet on the transparent object segmentation task. For convenience, we re-organize the annotation of the Trans10K[79] dataset from instance-level to object-level for training. The visual results shown in Fig. 11(c) further demonstrate the learning ability of †DGNet.

## 6 Conclusions

We presented a novel deep gradient learning framework (DGNet) for efficiently segmenting camouflaged objects. To extract the camouflaged features, we proposed to decouple the task into two branches, a context encoder and a texture encoder. We designed a novel plug-and-play module called gradient-induced transition (GIT), acting as a soft grouping module to learn features from these two branches jointly. This simple and flexible architecture showed strong generalization capabilities on three challenging datasets compared to the 20 SOTA competitors. In addition, our efficient version DGNet-S (8.3 M & 80 fps) achieved an excellent performance-efficiency trade-off. Our solution also produced visually ap-



(a) Polyp segmentation (b) Road crack detection (c) Transparent object segmentation

Fig. 11 Visualization results of three downstream applications. From top to bottom: input image (1st row), ground-truth (2nd row), and prediction (3rd row).

pealing results for three further applications, including polyp segmentation, defect detection, and transparent object segmentation, which validates its practical application value.

## Acknowledgements

The authors would like to thank the anonymous reviewers and editor for their helpful comments on this manuscript.

## Conflicts of Interests

The authors declared that they have no conflicts of interest in this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

## Open Access



## References

[1] D. P. Fan, G. P. Ji, G. L. Sun, M. M. Cheng, J. B. Shen, L. Shao. Camouflaged object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 2774–2784, 2020. DOI: 10.1109/CVPR42600.2020.00285.

[2] D. P. Fan, G. P. Ji, M. M. Cheng, L. Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to be published. DOI: 10.1109/TPAMI.2021.3085766.

[3] D. P. Fan, G. P. Ji, T. Zhou, G. Chen, H. Fu, J. Shen, L. Shao. PraNet: Parallel reverse attention network for polyp segmentation. In *Proceedings of the 23rd International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, Lima, Peru, pp. 263–273, 2020. DOI: 10.1007/978-3-030-59725-2_26.

[4] G. P. Ji, Y. C. Chou, D. P. Fan, G. Chen, H. Z. Fu, D. Jha, L. Shao. Progressively normalized self-attention network for video polyp segmentation. In *Proceedings of the 24th International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, Strasbourg, France, pp. 142–152, 2021. DOI: 10.1007/978-3-030-87193-2_14.

[5] G. P. Ji, G. Xiao, Y. C. Chou, D. P. Fan, K. Zhao, G. Chen, L. Van Gool. Video polyp segmentation: A deep learning perspective.Machine Intelligence Research, to be published. DOI: 10.1007/s11633-022-1371-y .

[6] D. P. Fan, T. Zhou, G. P. Ji, Y. Zhou, G. Chen, H. Z. Fu, J. B. Shen, L. Shao. Inf-Net: Automatic COVID-19 lung infection segmentation from CT images. *IEEE Transactions on Medical Imaging*, vol. 39, no. 8, pp. 2626–2637, 2020. DOI: 10.1109/TMI.2020.2996645.

[7] Y. H. Wu, S. H. Gao, J. Mei, J. Xu, D. P. Fan, R. G. Zhang, M. M. Cheng. JCS: An explainable COVID-19 diagnosis system by joint classification and segmentation. *IEEE Transactions on Image Processing*, vol. 30, pp. 3113–3126, 2021. DOI: 10.1109/TIP.2021.3058783.

[8] J. N. Liu, B. Dong, S. Wang, H. Cui, D. P. Fan, J. Q. Ma, G. Chen. COVID-19 lung infection segmentation with a novel two-stage cross-domain transfer learning framework. *Medical Image Analysis*, vol. 74, Article number 102205, 2021. DOI: 10.1016/j.media.2021.102205.

[9] G. P. Ji, K. R. Fu, Z. Wu, D. P. Fan, J. B. Shen, L. Shao. Full-duplex strategy for video object segmentation. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 4902–4913, 2021. DOI: 10.1109/ICCV48922.2021.00488.

[10] W. C. Chen, X. Y. Yu, L. L. Ou. Pedestrian attribute re-

cognition in video surveillance scenarios based on view-attribute attention localization. *Machine Intelligence Research*, vol. 19, no. 2, pp. 153–168, 2022. DOI: 10.1007/s11633-022-1321-8.

[11] J. R. Xue, J. W. Fang, P. Zhang. A survey of scene understanding by event reasoning in autonomous driving. *International Journal of Automation and Computing*, vol. 15, no. 3, pp. 249–266, 2018. DOI: 10.1007/s11633-018-1126-y.

[12] R. R. Feng, B. Prabhakaran. Facilitating fashion camouflage art. In *Proceedings of the 21st ACM International Conference on Multimedia*, ACM, Barcelona, Spain, pp. 793–802, 2013. DOI: 10.1145/2502081.2502121.

[13] M. Dean, R. Harwood, C. Kasari. The art of camouflage: Gender differences in the social behaviors of girls and boys with autism spectrum disorder,. *Autism*, vol. 21, no. 6, pp. 678–689, 2017. DOI: 10.1177/1362361316671845.

[14] H. Y. Mei, G. P. Ji, Z. Q. Wei, X. Yang, X. P. Wei, D. P. Fan. Camouflaged object segmentation with distraction mining. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, USA, pp. 8768–8777, 2021. DOI: 10.1109/CVPR46437.2021.00866.

[15] Y. Q. Lv, J. Zhang, Y. C. Dai, A. X. Li, B. W. Liu, N. Barnes, D. P. Fan. Simultaneously localize, segment and rank the camouflaged objects. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 11586–11596, 2021. DOI: 10.1109/CVPR46437.2021.01142.

[16] Q. Jia, S. L. Yao, Y. Liu, X. Fan, R. S. Liu, Z. X. Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, New Orleans, USA, pp. 4713–4722, 2022.

[17] Y. J. Zhong, B. Li, L. Tang, S. Y. Kuang, S. Wu, S. H. Ding. Detecting camouflaged object in frequency domain. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, USA, pp. 4504–4513, 2022.

[18] Q. Zhai, X. Li, F. Yang, C. Chen, H. Cheng, D. P. Fan. Mutual graph learning for camouflaged object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 12992–13002, 2021. DOI: 10.1109/CVPR46437.2021.01280.

[19] G. P. Ji, L. Zhu, M. C. Zhuge, K. R. Fu. Fast camouflaged object detection via edge-based reversible re-calibration network. *Pattern Recognition*, vol. 123, Article number 108414, 2022. DOI: 10.1016/j.patcog.2021.108414.

[20] H. W. Zhu, P. Li, H. R. Xie, X. F. Yan, D. Liang, D. P. Chen, M. Q. Wei, J. Qin. I can find you! Boundary-guided separated attention network for camouflaged object detection. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, online, pp. 3608–3616, 2022. DOI: 10.1609/aaai.v36i3.20273.

[21] F. Yang, Q. Zhai, X. Li, R. Huang, A. Luo, H. Cheng, D. P. Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 4126–4135, 2021. DOI: 10.1109/ICCV48922.2021.00411.

[22] A. X. Li, J. Zhang, Y. Q. Lv, B. W. Liu, T. Zhang, Y. C. Dai. Uncertainty-aware joint salient object and camouflaged object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 10066–10076, 2021. DOI: 10.1109/CVPR46437.2021.00994.

[23] Y. X. Pan, Y. W. Chen, Q. Fu, P. Zhang, X. Xu. Study on the camouflaged target detection method based on 3D convexity. *Modern Applied Science*, vol. 5, no. 4, pp. 152–157, 2011. DOI: 10.5539/mas.v5n4p152.

[24] P. Sengottuvelan, A. Wahi, A. Shanmugam. Performance of decamouflaging through exploratory image analysis. In *Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology*, IEEE, Nagpur, India, pp. 6–10, 2008. DOI: 10.1109/ICETET.2008.232.

[25] Z. Liu, K. Q. Huang, T. N. Tan. Foreground object detection using top-down information based on em framework. *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4204–4217, 2012. DOI: 10.1109/TIP.2012.2200492.

[26] J. Q. Yin, Y. B. Han, W. D. Hou, J. P. Li. Detection of the mobile object with camouflage color under dynamic background based on optical flow. *Procedia Engineering*, vol. 15, pp. 2201–2205, 2011. DOI: 10.1016/j.proeng.2011.08.412.

[27] J. Gallego, P. Bertolino. Foreground object segmentation for moving camera sequences based on foreground-background probabilistic models and prior probability maps. In *Proceedings of IEEE International Conference on Image Processing*, Paris, France, pp. 3312–3316, 2014. DOI: 10.1109/ICIP.2014.7025670.

[28] Y. J. Sun, G. Chen, T. Zhou, Y. Zhang, N. Liu. Context-aware cross-level fusion network for camouflaged object detection. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, Montreal, Canada, pp. 1025–1031, 2021. DOI: 10.24963/ijcai.2021/142.

[29] G. Chen, S. J. Liu, Y. J. Sun, G. P. Ji, Y. F. Wu, T. Zhou. Camouflaged object detection via context-aware cross-level fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, to be published. DOI: 10.1109/TCSVT.2022.3178173.

[30] J. J. Ren, X. W. Hu, L. Zhu, X. M. Xu, Y. Y. Xu, W. M. Wang, Z. J. Deng, P. A. Heng. Deep texture-aware features for camouflaged object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, to be published. DOI: 10.1109/TCSVT.2021.3126591.

[31] N. Kajiura, H. Liu, S. Satoh. Improving camouflaged object detection with the uncertainty of pseudo-edge labels. In *Proceedings of ACM Multimedia Asia*, Gold Coast, Australia, pp. 7, 2021. DOI: 10.1145/3469877.3490587.

[32] M. C. Zhuge, X. K. Lu, Y. Y. Guo, Z. H. Cai, S. H. Chen. CubeNet: X-shape connection for camouflaged object detection. *Pattern Recognition*, vol. 127, Article number 108644, 2022. DOI: 10.1016/j.patcog.2022.108644.

[33] T. N. Le, T. V. Nguyen, Z. L. Nie, M. T. Tran, A. Sugimoto. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, vol. 184, pp. 45–56, 2019. DOI: 10.1016/j.cviu.2019.04.006.

[34] Y. Pang, X. Zhao, T. Z. Xiang, L. Zhang, H. Lu. Zoom in and out: A mixed-scale triplet network for camouflaged object detection. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, USA, pp. 2160–2170, 2022.

[35] Y. X. Mao, J. Zhang, Z. X. Wan, Y. C. Dai, A. X. Li, Y. Q. Lv, X. Y. Tian, D. P. Fan, N. Barnes. Transformer transforms salient object detection and camouflaged object detection, [Online], Available: https://arxiv.org/abs/2104.

10127v1, 2021.

[36] X. L. Cheng, H. Xiong, D. P. Fan, Y. R. Zhong, M. Harandi, T. Drummond, Z. Y. Ge. Implicit motion handling for video camouflaged object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, New Orleans, USA, pp. 13864–13873, 2022.

[37] J. C. Zhu, X. Y. Zhang, S. Zhang, J. N. Liu. Inferring camouflaged objects by texture-aware interactive guidance network. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3599–3607, 2021.

[38] T. Y. Lin, P. Dollàr, R. Girshick, K. M. He, B. Hariharan, S. Belongie. Feature pyramid networks for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 936–944, 2017. DOI: 10.1109/CVPR.2017.106.

[39] Z. H. Ke, J. Y. Sun, K. C. Li, Q. Yan, R. W. H. Lau. MODNet: Real-time trimap-free portrait matting via objective decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, pp. 1140–1147, 2022. DOI: 10.1609/aaai.v36i1.19999.

[40] M. X. Tan, Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, USA, pp. 6105–6114, 2019.

[41] S. Ioffe, C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, pp. 448–456, 2015.

[42] X. Glorot, A. Bordes, Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, USA, pp. 315–323, 2011.

[43] Z. Su, W. Z. Liu, Z. T. Yu, D. W. Hu, Q. Liao, Q. Tian, M. Pietikäinen, L. Liu. Pixel difference networks for efficient edge detection. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 5097–5107, 2021. DOI: 10.1109/ICCV48922.2021.00507.

[44] C. Ma, Y. M. Rao, Y. A. Cheng, C. Chen, J. W. Lu, J. Zhou. Structure-preserving super resolution with gradient guidance. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 7766–7775, 2020. DOI: 10.1109/CVPR42600.2020.00779.

[45] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.

[46] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. DOI: 10.1109/TPAMI.2017.2699184.

[47] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[48] J. Wei, S. H. Wang, Q. M. Huang. F.3Net: Fusion, feedback and focus for salient object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12321–12328, 2020. DOI: 10.1609/aaai.v34i07.6916.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. M. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. J. Bai, S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Vancouver, Canada, pp. 721, 2019.

[50] S. M. Hu, D. Liang, G. Y. Yang, G. W. Yang, W. Y. Zhou. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences*, vol. 63, no. 12, Article number 222103, 2020. DOI: 10.1007/s11432-020-3097-4.

[51] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imageNet classification. In *Proceedings of IEEE International Conference on Computer Vision*, Santiago, Chile, pp. 1026–1034, 2015. DOI: 10.1109/ICCV.2015.123.

[52] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Lake Tahoe, USA, pp. 1097–1105, 2012.

[53] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, USA, 2015.

[54] I. Loshchilov, F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[55] P. Krähenbühl, V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Granada, Spain, pp. 109–117, 2011.

[56] J. X. Zhao, J. J. Liu, D. P. Fan, Y. Cao, J. F. Yang, M. M. Cheng. EGNet: Edge guidance network for salient object detection. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 8778–8787, 2019. DOI: 10.1109/ICCV.2019.00887.

[57] Z. Wu, L. Su, Q. M. Huang. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 7263–7272, 2019. DOI: 10.1109/ICCV.2019.00736.

[58] Z. Wu, L. Su, Q. M. Huang. Cascaded partial decoder for fast and accurate salient object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 3902–3911, 2019. DOI: 10.1109/CVPR.2019.00403.

[59] S. H. Gao, Y. Q. Tan, M. M. Cheng, C. Z. Lu, Y. P. Chen, S. C. Yan. Highly efficient salient object detection with 100K parameters. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 702–721, 2020. DOI: 10.1007/978-3-030-58539-6_42.

[60] J. Zhang, D. P. Fan, Y. C. Dai, S. Anwar, F. S. Saleh, T. Zhang, N. Barnes. UC-Net: Uncertainty inspired RGB-D saliency detection via conditional variational autoencoders. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle,

USA, pp. 8579–8588, 2020. DOI: 10.1109/CVPR42600.2020.00861.

[61] H. J. Zhou, X. H. Xie, J. H. Lai, Z. X. Chen, L. X. Yang. Interactive two-stream decoder for accurate and fast saliency detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 9138–9147, 2020. DOI: 10.1109/CVPR42600.2020.00916.

[62] Y. W. Pang, X. Q. Zhao, L. H. Zhang, H. C. Lu. Multi-scale interactive network for salient object detection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 9410–9419, 2020. DOI: 10.1109/CVPR42600.2020.00943.

[63] X. B. Qin, D. P. Fan, C. Y. Huang, C. Diagne, Z. C. Zhang, A. C. Sant'Anna, A. Suàrez, M. Jagersand, L. Shao. Boundary-aware segmentation network for mobile and web applications, [Online], Available: https://arxiv.org/abs/2101.04704, 2021.

[64] D. P. Fan, M. M. Cheng, Y. Liu, T. Li, A. Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of IEEE International Conference on Computer Vision*, Venice, Italy, pp. 4558–4567, 2017. DOI: 10.1109/ICCV.2017.487.

[65] D. P. Fan, C. Gong, Y. Cao, B. Ren, M. M. Cheng, A. Borji. Enhanced-alignment measure for binary foreground map evaluation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 698–704, 2018. DOI: 10.24963/ijcai.2018/97.

[66] D. P. Fan, G. P. Ji, X. B. Qin, M. M. Cheng. Cognitive vision inspired object segmentation metric and loss function. *SCIENTIA SINICA Informationis*, vol. 51, no. 9, Article number 1475, 2021. DOI: 10.1360/SSI-2020-0370.

[67] A. Borji, M. M. Cheng, H. Z. Jiang, J. Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015. DOI: 10.1109/TIP.2015.2487833.

[68] M. C. Zhuge, D. P. Fan, N. Liu, D. W. Zhang, D. Xu, L. Shao. Salient object detection via integrity learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to be published. DOI: 10.1109/TPAMI.2022.3179526.

[69] R. Margolin, L. Zelnik-Manor, A. Tal. How to evaluate foreground maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, USA, pp. 248–255, 2014. DOI: 10.1109/CVPR.2014.39.

[70] A. Howard, M. Sandler, B. Chen, W. J. Wang, L. C. Chen, M. X. Tan, G. Chu, V. Vasudevan, Y. K. Zhu, R. M. Pang, H. Adam, Q. Le. Searching for MobileNetV3. In *Proceedings of IEEE/CVF International conference on computer vision*, IEEE, Seoul, Republic of Korea, pp. 1314–1324, 2019. DOI: 10.1109/ICCV.2019.00140.

[71] D. Jha, P. H. Smedsrud, M. A. Riegler, P. Halvorsen, T. De Lange, D. Johansen, H. D. Johansen. Kvasir-SEG: A segmented polyp dataset. In *Proceedings of the 26th International Conference on Multimedia Modeling*, Springer, Daejeon, Republic of Korea, pp. 451–462, 2020. DOI: 10.1007/978-3-030-37734-2.

[72] J. Bernal, F. J. Sànchez, G. Fernàndez-Esparrach, D. Gil, C. Rodríguez, F. Vilariño. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. Saliency maps from physicians. *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015. DOI: 10.1016/j.compmedimag.2015.02.007.

[73] J. Bernal, J. Sànchez, F. Vilariño. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognition*, vol. 45, no. 9, pp. 3166–3182, 2012. DOI: 10.1016/j.patcog.2012.03.002.

[74] J. Silva, A. Histace, O. Romain, X. Dray, B. Granado. Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer. *International Journal of Computer Assisted Radiology and Surgery*, vol. 9, no. 2, pp. 283–293, 2014. DOI: 10.1007/s11548-013-0926-3.

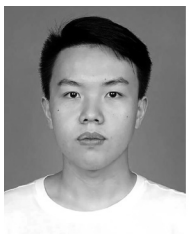
[75] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, Munich, Germany, pp. 234–241, 2015. DOI: 10.1007/978-3-319-24574-4_28.

[76] Z. W. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. M. Liang. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020. DOI: 10.1109/TMI.2019.2959609.

[77] X. Q. Zhao, L. H. Zhang, H. C. Lu. Automatic polyp segmentation via multi-scale subtraction network. In *Proceedings of the 24th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, Strasbourg, France, pp. 120–130, 2021. DOI: 10.1007/978-3-030-87193-2_12.

[78] Y. Shi, L. M. Cui, Z. Q. Qi, F. Meng, Z. S. Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016. DOI: 10.1109/TITS.2016.2552248.

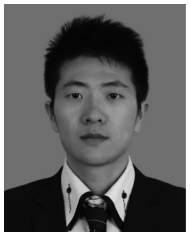
[79] E. Z. Xie, W. J. Wang, W. H. Wang, M. Y. Ding, C. H. Shen, P. Luo. Segmenting transparent objects in the wild. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 696–711, 2020. DOI: 10.1007/978-3-030-58601-0_41.



**Ge-Peng Ji** received the M. Sc. degree in communication and information systems from Wuhan University, China in 2021. He is a Ph.D. student at Australian National University, supervised by Professor Nick Barnes, majoring in engineering and computer science. He has published about 10 peer-reviewed journal and conference papers. In 2021, he received the Student Travel Award from Medical Image Computing and Computer-Assisted Intervention Society.

His research interests lie in computer vision, especially in a variety of dense prediction tasks, such as video analysis, medical image segmentation, camouflaged object segmentation, and saliency detection.

E-mail: gepengai.ji@gmail.com
ORCID iD: 0000-0001-7092-2877



**Deng-Ping Fan** received the Ph. D. degree from Nankai University, China in 2019. He joined the Inception Institute of Artificial Intelligence (IIAI), UAE in 2019. He is a Postdoctoral Researcher, working with Prof. Luc Van Gool in Computer Vision Laboratory, ETH Zürich, Switzerland. He has published approximately 50 top journal and conference papers such as TPAMI, CVPR, ICCV, ECCV, etc. He won the Best Paper Fi-

nalist Award at IEEE CVPR 2019, and the Best Paper Award Nominee at IEEE CVPR 2020. He was recognized as the CVPR 2019 outstanding reviewer with a special mention award, the CVPR 2020 outstanding reviewer, the ECCV 2020 high-quality reviewer, and the CVPR 2021 outstanding reviewer. He served as a program committee board (PCB) member of IJCAI 2022–2024, a senior program committee (SPC) member of IJCAI 2021, a committee member of China Society of Image and Graphics (CSIG), area chair in NeurIPS 2021 Datasets and Benchmarks Track, area chair in MICCAI2020 Workshop (OMIA7), editorial board member of *Computer Vision and Machine Learning*.

His research interests include computer vision, deep learning, and visual attention, especially the human vision on co-salient object detection, RGB salient object detection, RGB-D salient object detection, and video salient object detection.

E-mail: dengpfan@gmail.com (Corresponding author)
ORCID iD: 0000-0002-5245-7518

**Yu-Cheng Chou** received the B. Sc. degree in software engineering from School of Computer Science, Wuhan University, China in 2022. He is currently a visiting student at Johns Hopkins University, supervised by Zongwei Zhou and Prof. Alan Yuille.

His research interests include medical imaging, causality, and computer vision, especially developing novel methodologies to detect lesions accurately and exploring explainability through causality for computer-aided diagnosis and surgery.

E-mail: johnson111788@gmail.com
ORCID iD: 0000-0002-9334-2899

**Dengxin Dai** received the Ph. D. degree in computer vision from ETH Zürich, Switzerland in 2016. He is a senior research group leader at the MPI for Informatics, heading the research group vision for autonomous systems. He has been area chair of multiple major computer vision conferences (e.g., CVPR21, CVPR22, ECCV22), has organized multiple international workshops, is on the editorial board of IJCV, and is an ELLIS member. His team has won multiple awards including the 1st Place at Waymo Open Dataset Challenge 2022 and the 2nd Place at NuScenes Tracking Challenge 2021. He has received the Golden Owl Award with ETH Zürich in 2021 for his exceptional teaching.

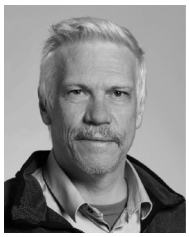
His research interests lie in autonomous driving, robust perception in adverse weather and illumination conditions, domain adaptation, sensor fusion, multi-task learning, and object recognition under limited supervision.

E-mail: dai@vision.ee.ethz.ch
ORCID iD: 0000-0001-5440-9678

**Alexander Liniger** received the B. Sc. and M. Sc. degrees in mechanical engineering from Department of Mechanical and Process Engineering, ETH Zürich, Switzerland in 2010 and 2013, respectively, and received the Ph. D. degree at Automatic Control Laboratory, ETH Zürich, Switzerland in 2018. Currently, he is a postdoctoral researcher in Computer Vision Laboratory, ETH Zürich, Switzerland, where he is part of Luc van Gool′s group working on the Toyota TRACE project.

During his Ph. D., his main research interests include model predictive control, viability theory as well as game theory and their application to autonomous driving and racing. Currently, he is investigating how control theory and computer vision can be combined to achieve end-to-end learning approaches with formal guarantees.

E-mail: alex.liniger@vision.ee.ethz.ch
ORCID iD: 0000-0002-7858-7900



**Luc Van Gool** received the Ph. D. degree in electromechanical engineering from Katholieke Universiteit Leuven, Belgium in 1981. Currently, he is a professor at the Katholieke Universiteit Leuven in Belgium and the ETH Zürich, Switzerland. He leads computer vision research at both places and also teaches at both. He has been a program committee member of several major computer vision conferences. He received several Best Paper awards, won a David Marr Prize and a Koenderink Award, and was nominated Distinguished Researcher by the IEEE Computer Science Committee. He is a co-founder of 10 spin-off companies.

His interests include 3D reconstruction and modeling, object recognition, tracking, gesture analysis, and a combination of those.

E-mail: vangool@vision.ee.ethz.ch
ORCID iD: 0000-0002-3445-5711