A Revisit to the Decoder for Camouflaged Object Detection

Seung Woo Ko*[1,4]
seungwoo.ko@lgresearch.ai
Joopyo Hong*[1]
dals2539@snu.ac.kr
Suyoung Kim*[2]
ksyo96@snu.ac.kr
Seungjai Bang*[3,5]
epdl6403@snu.ac.kr
Sungzoon Cho[3]
zoon@snu.ac.kr
Nojun Kwak[2]
nojunk@snu.ac.kr
Hyung-Sin Kim†[1]
hyungkim@snu.ac.kr
Joonseok Lee†[1,6]
joonseok@snu.ac.kr

[1] Graduate School of Data Science, Seoul National University Seoul, Korea
[2] Department of Intelligence and Information, Seoul National University Seoul, Korea
[3] Department of Industrial Engineering, Seoul National University Seoul, Korea
[4] LG AI Research Seoul, Korea
[5] LG Electronics Seoul, Korea
[6] Google Research Mountain View, CA, USA
*Equal contribution
†Corresponding authors

Abstract

Camouflaged object detection (COD) aims to generate a fine-grained segmentation map of camouflaged objects hidden in their background. Due to the hidden nature of camouflaged objects, it is essential for the decoder to be tailored to effectively extract proper features of camouflaged objects and extra-carefully generate their complex boundaries. In this paper, we propose a novel architecture that augments the prevalent decoding strategy in COD with Enrich Decoder and Retouch Decoder, which help to generate a fine-grained segmentation map. Specifically, the Enrich Decoder amplifies the channels of features that are important for COD using channel-wise attention. Retouch Decoder further refines the segmentation maps by spatially attending to important pixels, such as the boundary regions. With extensive experiments, we demonstrate that ENTO shows superior performance using various encoders, with the two novel components playing their unique roles that are mutually complementary.

1 Introduction

Object segmentation is a widely researched topic in computer vision. Of its several branches, Camouflaged Object Detection (COD) [5] targets images that contain naturally or

Figure 1: **Examples of camouflaged objects in COD10K and NC4K.**

Methods	COD10K (2,026)				NC4K (4,121)				CAMO (250)			
	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
ZoomNet [25]	**0.870**	0.782	0.912	0.023	**0.884**	0.829	**0.922**	**0.034**	**0.865**	**0.812**	**0.914**	**0.052**
HitNet [12]	0.867	**0.803**	**0.926**	**0.022**	0.872	**0.832**	0.921	0.037	0.836	0.798	0.893	0.060

Table 1: **Comparison of baseline models using Transformer encoder.** ZoomNet with the same encoder setting as HitNet shows comparable or even better results for some metrics.

artificially camouflaged objects and aims to correctly *segment*[1] them from the background. COD is applied to polyp segmentation [6] in medicine, surface defect detection in manufacturing [1], enemy detection in military [18], and camouflaged organisms in ecology [24].

Camouflaged objects bring a variety of challenges in segmentation. Fig. 1 illustrates several representative examples: using protective coloring to disturb others' visual recognition, hiding behind another object (occlusion), or exhibiting complex shapes. To segment these objects accurately, the model needs not only to precisely understand the given scene and individual objects, but also to be equipped with an exceptional segmentation map generator, surpassing the level of general object segmentation models.

Numerous methods have been proposed for this task, primarily focusing on the extraction of rich features to distinguish inconspicuous objects from their surroundings [12, 13, 19]. It has been demonstrated that using high-resolution input images is effective for COD [12, 34], more than that for general object segmentation [9, 26, 35]. However, has the performance improvements using such powerful encoding tools been matched with an equally effective decoding strategy? To verify this, we replace the CNN encoder backbone in ZoomNet [25] with a Transformer backbone, HitNet [12]. The results in Tab. 1 indicates that the older ZoomNet easily reaches or even surpasses the performance achieved by HitNet by simply taking its Transformer-based encoder and higher resolution input. This implies that there has been little advances in the decoding architecture, and there is room for improvement in the decoder structure to fully take advantage of the encoding capacity advanced by recent works.

In this regards, we propose to equip the base decoder [7] with two additional processes, before and after it. Before the base decoder, we *enrich the image features* so they are more suitable for camouflaged objects, since the regular image features extracted by the encoder may not be sufficiently informative. Based on the enriched features and a coarse prediction map from this pre-decoding step, the base decoder generates a refined segmentation map. After the decoding, the segmentation map is *retouched again* since it may still not be perfect, especially regarding the complex shapes of camouflaged objects.

Reflecting this idea, we propose **ENTO**, equipped with the **EN**rich and Re**TO**uch De-

[1]Precisely speaking, this is an image segmentation task, where the expected output is a pixel-wise segmentation map, not bounding boxes. We use this idiomatic expression interchangeably.

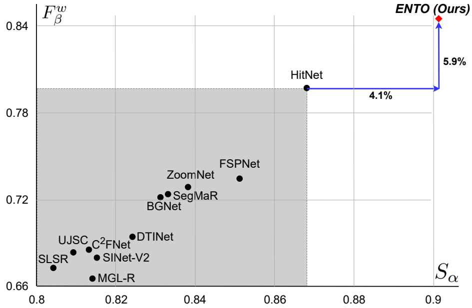

Figure 2: **Performances in two representative metrics, S_α and F_β^w, on COD10K [5].**

coders that sandwich the base decoder, specially designed for camouflaged object segmentation. The core in ENTO is a step-wise generation of the segmentation map through three consecutive decoding steps. Specifically, the first step (Enrich Decoder) is to reorganize the image features by channel-wise attention and combination of multi-level features, focusing on the features (*e.g.*, texture and shape) critical for the COD task. The enriched features are then fed into the base decoder [7] to output segmentation maps at different scales. The last Retouch Decoder further enhances the boundaries of the objects by applying spatial attention to focus on those regions of the segmentation map. These pre- and post-steps enable more precise representation of object boundaries while keeping the overall structure.

Our comprehensive experiments verify that ENTO achieves state-of-the-art performance across multiple representative COD datasets, as in Fig. 2. We also show through extensive experiments that the proposed decoder structure is able to adapt to any type of feature encoder at any image resolution, surpassing previous state-of-the-art models.

2 Related Work

Camouflaged Object Detection (COD). The COD task was first proposed as a sub-task of object segmentation by Fan *et al.* [5], providing a large-scale dataset (COD10K). Since then, numerous models have been proposed to tackle this problem [2, 7, 12, 13, 15, 21, 39]. Since camouflaged objects are harder to segment, auxiliary tasks are often applied, such as object ranking [21] and edge detection [2]. Some works are inspired by biology. SINet [5], for example, takes an idea from predators that first search for prey in a general sense and subsequently identify its precise location. BSA-Net [39] mimics how human discovers camouflaged objects using a two-stream attention network.

Input and Encoding. Recently, high resolution images have been utilized for COD, taking advantage of richer information on the boundary regions. HitNet [12] utilizes high resolution images to better capture high-frequency details. SARNet [34] improves the segmentation quality by training on higher resolution images.

To extract fine-grained features from intricate images, Transformers [31] have been adopted in COD. Pyramid Vision Transformer (PVT) [32] improves the vanilla ViT [3] by providing a pyramid-style encoding structure, extracting features in a similar manner to CNNs. Recent COD models adopt ViT [13] and PVT [12, 34] as the feature extractor.

Decoding Strategy. The U-Net architecture [29] has been extensively utilized for intricate object segmentation domains, such as medical imaging and remote sensing. Drawing inspiration from SINet [5], a majority of studies on COD are rooted in this U-shaped decoding, sequentially aggregating high to low-level features with skip connections. Some recent works have tried to enhance decoding strategies for COD. PreyNet [37] densely aggregates features from neighboring layers during the high-to-low decoding phase. HitNet [12] employs a feedback refinement between layers. FSPNet [13] employs a pyramidal shrinkage decoding strategy that progressively aggregates adjacent features. Furthermore, several studies [23, 39] employ a coarse-to-fine decoding strategy, initially generating a coarse map followed by refinement stages that aim to capture information grounded in the coarse map.

Despite studies on decoding strategies, they remain relatively unexplored compared to the recent significant advances in encoders. In particular, while the performance of decoders based on CNN-based encoders is almost saturated, they fall short in effectively leveraging the rich feature maps produced by advanced Transformer-based encoders like ViT [13] and PVT [12, 34]. In response to this, we introduce an innovative decoding strategy that augments the prevalent decoding structure with preprocessing and postprocessing decoders that help to enhance the features necessary for COD and improve the boundary details.

3 Preliminary

Problem Formulation and Notations. The Camouflaged Object Detection (COD) is formulated in the same way as a regular image segmentation problem, except that the target object is camouflaged and not easily seen at a glance. The input is an image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ with RGB channels, where H and W stand for the height and width of the image. For each image, the COD model needs to generate a bitmap $\hat{\mathbf{Y}} \in \{0,1\}^{H \times W}$, predicting the ground truth map $\mathbf{Y} \in \{0,1\}^{H \times W}$, where 1 indicates the pixel belongs to the camouflaged object and 0 otherwise. For simplicity, we do not distinguish the identity of an object, even if there are multiple camouflaged objects in the same scene.

Our Base Architecture. We take a common encoder-decoder structure for image segmentation, where the encoder extracts essential features to segment the target object, and the decoder generates a segmentation map corresponding to it. We adopt a particular encoder and decoder architecture described below, which is widely used for modern COD models.

Feature Encoder. Given an input image $\mathbf{X}$, the encoder E extracts features $\mathbf{f}_i$ with different resolutions at each level i, where $i = 1, ..., L$. E may accommodate various types of backbones, *e.g.*, CNNs, or Transformers. As most image encoders generate different numbers of channels at each level, we make the channel size the same across the layers by additionally applying a convolution layer. We denote the resulting feature map by $\mathbf{f}'_i$, where $i = 1, ..., L$.

Base Decoder. In this work, we propose pre- and post-decoding steps on an existing decoding structure, which we call base decoder. Aligned with the encoder, we adopt a common multi-level decoder to fully take advantage of the multi-level feature maps. As illustrated in Fig. 3, we take L levels of Group Attention Blocks (GABs) inspired by [7]. GAB is a residual learning process employing the group guidance operation, focusing more on important information about objects through attention with guidance from the prior segmentation map and gradually improving the map through a sequence of S group attentions. It expands the segmentation map step by step from low to high resolution, learning rough and abstract

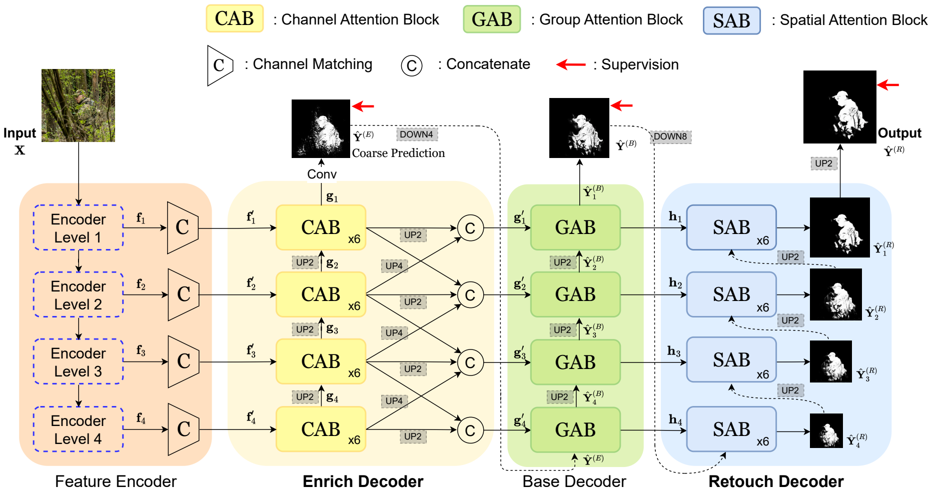

Figure 3: **Overall Architecture of ENTO, comprising a feature encoder and the three consecutive decoders.** We show the architecture with 4 levels ($L = 4$), consistent with our full model, but the number of levels L can be adjusted according to the feature encoder.

patterns at the low resolution while finer details at the higher resolution. The operations of GAB are detailed in Appendix A. Although such a multi-level decoder structure is prevalent in COD, our base decoder slightly deviates from [7] in three ways: 1) we remove the reverse operation on the guidance maps, 2) we expand each GAB to take four group operations, and 3) we provide an additional GAB stage to fully utilize the features from the encoder.

4 The Proposed Method: ENTO

On top of the base encoder-decoder structure in Sec. 3, we introduce two novel steps that make the segmentation model more suitable for camouflaged objects, as illustrated in Fig. 3. Specifically, we insert a pre-decoding step (Enrich Decoder) between the encoder and the base decoder to adapt the encoded image features more suitable for COD, and a post-decoding step (Retouch Decoder) at the end, to refine the produced segmentation map, focusing on the object boundaries.

The overall decoding structure is as follows. Taking the encoded image features at L different scales, Enrich Decoderselectively amplifies the channels that contain important cues for detecting camouflaged objects. Taking an attention strategy used for image super-resolution, this step benefits to prepare a particular set of features (*e.g.*, texture and shape) that are more important for camouflaged objects. Additionally, features at different resolutions are fused so that the base decoder can utilize both coarse and fine information simultaneously at each level. Then, the base decoder adds details step-by-step, refining the low-resolution segmentation map using higher resolution features produced by the previous step. We use Group Attention Blocks (GABs) [7], suitable to handle our multi-resolution features, but any image decoder can be used for this step. Lastly, the Retouch Decoder further enhances the object boundaries of the produced segmentation map. We adopt spatial

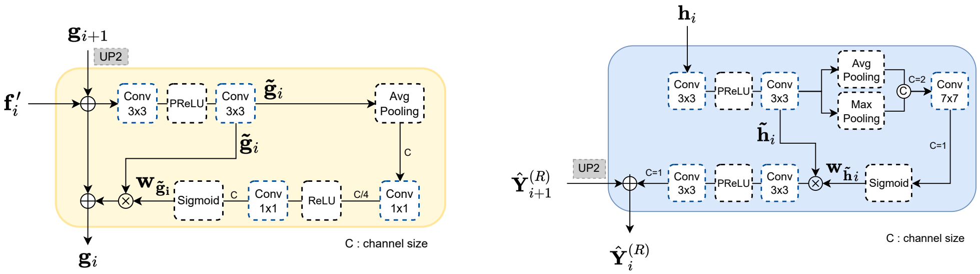

Figure 4: **Channel Attention Block (CAB) [38] and Spatial Attention Block (SAB).**

attention to amplify the signals of the edges and detailed areas that may not have been fully segmented by the base decoder.

4.1 Pre-Decoding Step: Enrich Decoder

In order to effectively utilize the encoded features $\mathbf{f}'_i$, we propose a novel pre-decoding step to adapt the features towards the camouflaged objects, before the base decoder. Specifically, we apply a module that has been shown effective in image super-resolution; namely, channel attention block (CAB) [38]. This module selectively emphasizes particular feature channels important for super-resolution by channel-wise attention to the image features. Such a mechanism would benefit the COD task similarly, since some different sets of features (*e.g.*, texture and shape) may be more important than others (*e.g.*, color) to detect camouflaged objects, unlike regular ones.

As illustrated in Fig. 4(left), the channel attention block (CAB) takes two inputs: the general feature map $\mathbf{f}'_i$ at the i-th level, and the adapted feature map $\mathbf{g}_{i+1}$ by the previous CAB. At each level i, CAB upsamples $\mathbf{g}_{i+1}$ (to match the dimensionality with $\mathbf{f}'_i$), adds them with $\mathbf{f}'_i$, takes channel-wise attention to reogranize the information, and residually adds the input features back to the output $\mathbf{g}_i$. Formally, CAB at level i performs:

$$\begin{aligned} &\tilde{\mathbf{g}}_i = \texttt{Conv3} \circ \texttt{PReLU} \circ \texttt{Conv3}(\mathbf{f}'_i \oplus \mathbf{g}_{i+1}), \quad \mathbf{w}_{\tilde{\mathbf{g}}_i} = \sigma[\texttt{Conv1} \circ \texttt{ReLU} \circ \texttt{Conv1}(\texttt{Pool}(\tilde{\mathbf{g}}_i))] \\ &\mathbf{g}_i = (\mathbf{w}_{\tilde{\mathbf{g}}_i} \otimes \tilde{\mathbf{g}}_i) \oplus (\mathbf{f}'_i \oplus \mathbf{g}_{i+1}) \end{aligned} \tag{1}$$

where $i = 1, \ldots, L$, and $\mathbf{w}_{\tilde{\mathbf{g}}_i} \in (0,1)^C$ is the channel attention weights. `Conv3`, `Conv1`, and `Pool` stand for 3×3 and 1×1 convolutions, and global average pooling, respectively. $\oplus$ and $\otimes$ indicate element-wise addition and channel-wise multiplication, respectively. At the highest level L, CAB simply performs channel-wise attention on $\mathbf{f}'_L$, taking only $\mathbf{f}'_L$ as input (that is, $\mathbf{g}_{L+1} = 0$).

Then, how does this CAB produce features more suitable for camouflage objects? At the lowest layer, Enrich Decoder produces a coarse segmentation map $\hat{\mathbf{Y}}^{(E)}$, as illustrated in Fig. 3, with an additional convolution. By applying a loss on the difference between this coarse segmentation map and the ground truth, we guide the Enrich Decoder to learn a set of feature maps suitable for the camouflage objects. We also use this coarse map as the starting point of the base decoder, so the loss applied in the subsequent decoders indirectly affects Enrich Decoder to learn COD-specific features as well. In this way, the features important to COD are selectively amplified.

Lastly, the Enrich Decoder concatenates the produced feature $\mathbf{g}_i$ with that on one level lower and higher ($\mathbf{g}_{i-1}$ and $\mathbf{g}_{i+1}$). These aggregated features are appropriately upsampled, concatenated, and reverted back to the original channel size using convolution to produce a single feature map $\mathbf{g}'_i$. This multi-level fusion allows the base decoder to access useful information scattered at different resolutions.

4.2 Post-Decoding Step: Retouch Decoder

In addition to the pre-decoding step, we also add a post-decoding step to further improve the quality of the segmentation map produced by the base decoder. Particularly, the Retouch Decoder performs fine-tuned enhancement, focusing on object boundaries, through spatial attention on segmentation maps starting from the final output from the base decoder. Similarly to other previous steps, Retouch Decoder progressively generates expanding prediction maps starting from the lowest-resolution.

Each Retouch Decoder at the i-th level applies multiple spatial attention blocks (SAB) that take the features $\mathbf{h}_i$ from the base decoder and the segmentation map from the previous SAB stage. As shown in Fig. 4(right), each SAB generates spatial weights $\mathbf{w}_{\tilde{\mathbf{h}}_i}$ indicating which pixels should be further modified from the previous map. Applying convolutions on the features weighted by this generates an improved map, added to the higher level map $\hat{\mathbf{Y}}^{(R)}_{i+1}$ to produce enhanced segmentation map $\hat{\mathbf{Y}}^{(R)}_i$. Overall, SAB at the i-th level operates

$$\begin{aligned} \tilde{\mathbf{h}}_i &= \texttt{Conv3} \circ \texttt{PReLU} \circ \texttt{Conv3}(\mathbf{h}_i), \quad \mathbf{w}_{\tilde{\mathbf{h}}_i} = \sigma(\texttt{Conv7} \circ \texttt{Pool}(\tilde{\mathbf{h}}_i)) \\ \hat{\mathbf{Y}}^{(R)}_i &= [\texttt{Conv3} \circ \texttt{PReLU} \circ \texttt{Conv3}(\mathbf{w}_{\tilde{\mathbf{h}}_i} \otimes \tilde{\mathbf{h}}_i)] \oplus \hat{\mathbf{Y}}^{(R)}_{i+1} \end{aligned} \tag{2}$$

where $i = 1, \ldots, L$ and $\hat{\mathbf{Y}}^{(R)}_{L+1} = \hat{\mathbf{Y}}^{(B)}$. $\tilde{\mathbf{h}}_i$ is the intermediate feature with the same size as $\mathbf{h}_i$, and $\mathbf{w}_{\tilde{h}_i} \in (0,1)^{H \times W}$ is the spatial attention weights. `Pool` indicates a concatenation of average and max pooled features channel-wise; that is, $[\texttt{AVG}_\texttt{c}(\tilde{\mathbf{h}}_i), \texttt{MAX}_\texttt{c}(\tilde{\mathbf{h}}_i)]$. Through these operations, SAB ultimately learns to identify the pixels that need modifications through spatial attention and provides residual learning to the output segmentation maps.

4.3 Training Objectives

ENTO sequentially refines a segmentation map throughout the decoding steps, supervising to the coarse map generated by the Enrich Decoder and to each of the final output maps from base decoder and Retouch Decoder. Fig. 3 illustrates where we apply supervision. In each case, the output map is first upscaled to the input resolution, then we apply the following two common loss functions for object detection: pixel-level weighted binary cross entropy (wBCE) [28] and weighted intersection-over-union (wIOU) loss [33] to account for the overall overlap of the output map with the ground truth. For both losses, each pixel is weighted to reflect its difficulty to be detected. The loss at each decoding step is given by

$$\mathcal{L}(\hat{\mathbf{Y}}, \mathbf{Y}) = \mathcal{L}_{\text{wBCE}}(\texttt{UP}(\hat{\mathbf{Y}}), \mathbf{Y}) + \mathcal{L}_{\text{wIOU}}(\texttt{UP}(\hat{\mathbf{Y}}), \mathbf{Y}) \tag{3}$$

and by combining them, the overall loss is given by

$$\mathcal{L} = \underbrace{\mathcal{L}(\hat{\mathbf{Y}}^{(E)}, \mathbf{Y})}_{\text{Enrich Decoder}} + \underbrace{\mathcal{L}(\hat{\mathbf{Y}}^{(B)}, \mathbf{Y})}_{\text{base decoder}} + \underbrace{\mathcal{L}(\hat{\mathbf{Y}}^{(R)}, \mathbf{Y})}_{\text{Retouch Decoder}}, \tag{4}$$

where $\hat{\mathbf{Y}}^{(E,B,R)}$ are output segmentation maps at each decoder, and $\mathbf{Y}$ is the ground truth segmentation map. `UP` is upscaling function to the input image size using bilinear interpolation.

Methods	Publications	COD10K (2,026)				NC4K (4,121)				CAMO (250)			
		$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
SINet [5]	CVPR_{20}	0.771	0.551	0.806	0.051	0.808	0.723	0.871	0.058	0.751	0.606	0.771	0.100
SLSR [20]	CVPR_{21}	0.804	0.673	0.880	0.037	0.840	0.766	0.895	0.048	0.787	0.696	0.838	0.080
MGL-R [36]	CVPR_{21}	0.814	0.666	0.852	0.035	0.833	0.740	0.867	0.052	0.775	0.673	0.812	0.088
PFNet [23]	CVPR_{21}	0.800	0.660	0.877	0.040	0.829	0.745	0.888	0.053	0.782	0.695	0.842	0.085
UJSC [17]	CVPR_{21}	0.809	0.684	0.884	0.035	0.842	0.771	0.898	0.047	0.800	0.728	0.859	0.073
C^2FNet [30]	IJCAI_{21}	0.813	0.686	0.890	0.036	0.838	0.762	0.897	0.049	0.796	0.719	0.854	0.080
SINet-V2 [7]	TPAMI_{22}	0.815	0.680	0.887	0.037	0.847	0.770	0.903	0.048	0.820	0.743	0.882	0.070
BGNet [2]	IJCAI_{22}	0.831	0.722	0.901	0.033	0.851	0.788	0.907	0.044	0.812	0.749	0.870	0.073
DTINet [19]	ICPR_{22}	0.824	0.695	0.896	0.034	0.863	0.792	**0.917**	**0.041**	**0.857**	0.796	**0.916**	**0.050**
SegMaR [14]	CVPR_{22}	0.833	0.724	0.899	0.034	0.841	0.781	0.896	0.046	0.815	0.753	0.874	0.071
ZoomNet [25]	CVPR_{22}	0.838	0.729	**0.911**	0.029	0.853	0.784	0.912	0.043	0.820	0.752	0.883	0.066
FEDER-R2N [10]	CVPR_{23}	0.844	-	**0.911**	0.029	0.862	-	0.913	0.042	0.836	-	0.897	0.066
FSPNet [13]	CVPR_{23}	**0.851**	**0.735**	0.895	**0.026**	**0.879**	**0.816**	0.915	0.048	**0.856**	**0.799**	0.899	**0.050**
HitNet [12]	AAAI_{23}	**0.868**	**0.798**	**0.932**	**0.024**	**0.870**	**0.825**	**0.921**	**0.039**	0.844	**0.801**	**0.902**	**0.057**
ENTO (Ours)	-	**0.904**	**0.845**	**0.948**	**0.018**	**0.904**	**0.864**	**0.942**	**0.029**	**0.881**	**0.841**	**0.928**	**0.047**

Table 2: **Overall comparison on COD datasets.** The **1st**, **2nd**, **3rd** best are highlighted.

5 Experiments

5.1 Experiment Setup

Datasets. We evaluate on three widely-used COD datasets: COD10K [5], CAMO [16] and NC4K [21]. COD10K includes 5,066 camouflaged, 3,000 background, and 1,934 non-camouflaged images. CAMO consists of 1,250 camouflaged and 1,250 non-camouflaged images. NC4K consists of 4,121 images containing camouflaged objects from the Internet. We train the model using only camouflaged images from COD10K and CAMO train sets (4,040 images) and evaluate on NC4K, COD10K, and CAMO test sets. For evaluation metrics, we use

Evaluation Metrics. We evaluate with four common metrics: S-measure (S_α) [4], weighted F-measure (F_β^w) [22], mean E-measure (E_ϕ) [8], and mean absolute error (M) [27]. S-measure quantifies the structural similarity between the model output and the ground truth, which is important in COD tasks that usually contain complex shapes of objects. Weighted F-measure is a modified version of F-measure that provides more reliable evaluation. Mean E-measure quantifies the pixel-level matching and image-level statistics between the predicted output and the ground truth. Mean absolute error directly quantifies the error in each pixel value averaged over the whole image. See Appendix C for more experimental settings.

5.2 Comparison to Existing Methods

Quantitative Comparison. As reported in Tab. 2, our proposed method outperforms all baseline methods in all metrics. In particular, on COD10K, ENTO shows of 4.15%, 5.89%, and 1.72% improvement in S_α, F_β^w, and E_ϕ from the next best model, HitNet [12]. Similarly, on NC4K, ENTO improves 2.84% in S_α and 4.73% in F_β^w from the second-best models. These results demonstrate that our model competently segments camouflaged objects.

Impartial Comparison. As a high-performance encoder and high-resolution input images evidently lead to better performance [12, 26, 34, 35], a fair comparison should be conducted with the baselines; *e.g.*, same encoder backbone and input image resolution. Tab. 3 compares the performance of our model with the best-performing baseline models under various com-

Encoder Backbone	Resolution	Best Baseline	Publications	$S_\alpha \uparrow$ Ours	$S_\alpha \uparrow$ Base	$F_\beta^w \uparrow$ Ours	$F_\beta^w \uparrow$ Base	$E_\phi \uparrow$ Ours	$E_\phi \uparrow$ Base	$M \downarrow$ Ours	$M \downarrow$ Base
PVTv2-B2 [32]	352×352	HitNet [12]	$AAAI_{23}$	**0.889**	0.827	**0.835**	0.727	**0.932**	0.907	0.032	**0.029**
	704×704	HitNet [12]	$AAAI_{23}$	**0.900**	0.870	**0.857**	0.825	**0.937**	0.921	**0.031**	0.039
ViT [3]	384×384	FSPNet [13]	$CVPR_{23}$	**0.886**	0.879	**0.834**	0.816	**0.935**	0.915	**0.033**	0.048
Res2Net50 [5]	352×352	SINet-V2 [7]	$PAMI_{22}$	**0.854**	0.847	**0.775**	0.770	0.901	**0.903**	**0.046**	0.048
	384×384	FEDER-R2N [10]	$CVPR_{23}$	**0.864**	0.862	0.802	–	**0.913**	**0.913**	**0.041**	0.042
ResNet50 [11]	576×576	ZoomNet [25]	$CVPR_{22}$	**0.858**	0.853	**0.786**	0.784	**0.914**	0.912	**0.043**	**0.043**

Table 3: **Comparison using same encoder and resolution settings as baselines on NC4K dataset.** We select the best baseline models using various encoder backbone and resolutions, and report our model's performance under that setting. The better result is boldfaced.

Methods	Publications	COD10K (2,026) $S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	NC4K (4,121) $S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$	CAMO (250) $S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
ZoomNet [25]	$CVPR_{22}$	0.870	0.782	0.912	0.023	0.884	0.829	0.922	0.034	0.865	0.812	0.914	0.052
HitNet [12]	$AAAI_{23}$	0.867	0.803	0.926	0.022	0.872	0.832	0.921	0.037	0.836	0.798	0.893	0.060
ENTO (Ours)	-	**0.894**	**0.827**	**0.944**	**0.021**	**0.895**	**0.849**	**0.935**	**0.033**	**0.882**	**0.837**	**0.928**	**0.046**

Table 4: **Comparison to baseline models using Transformer encoder.**

binations of encoder backbones and resolutions. Our innovative decoding strategy achieves the best performance in most metrics with the same backbone and resolutions as baselines.

In Tab. 4, we compare our model with the same Transformer backbone (PVTv2-b2) and input resolution (768), similar to the results shown in Tab. 1. ENTO significantly outperforms both models using Transformer backbone showing that it matches the advances in encoder performance, such as using Transformer-base backbones, while the previous models have not fully utilized them.

Qualitative Comparison. Fig. 5 illustrates a few examples on COD10K. Our model successfully captures both the overall structure and the fine details, such as slim legs or complex edges, compared to other models. Additionally, ENTO captures small objects accurately, and at the same time, accurately covers the entirety of the object for big objects. In spite of occlusion, ENTO accurately captures the target object, while other models often include such occluded parts. Even for objects with complex shapes, our model is able to capture fine details, while other models tend to inaccurately capture such details.

5.3 Ablation Study & Visualization

We evaluate the effects of the proposed components by measuring the performance improvement over the base decoder alone. Tab. 5 verifies on COD10K that adding each component improves the performance. As expected, the best performance is achieved with both components, showing that they are complementary. In Fig. 6, we illustrate that adding the

Decoders	$S_\alpha \uparrow$	$F_\beta^w \uparrow$	$E_\phi \uparrow$	$M \downarrow$
Base	0.890	0.815	0.938	0.021
Base + Enrich	0.899	0.836	0.946	0.019
Base + Retouch	0.896	0.825	0.935	0.020
ENTO (ours)	**0.904**	**0.845**	**0.948**	**0.018**

Table 5: **Ablation study on decoding structure of ENTO on COD10K.**

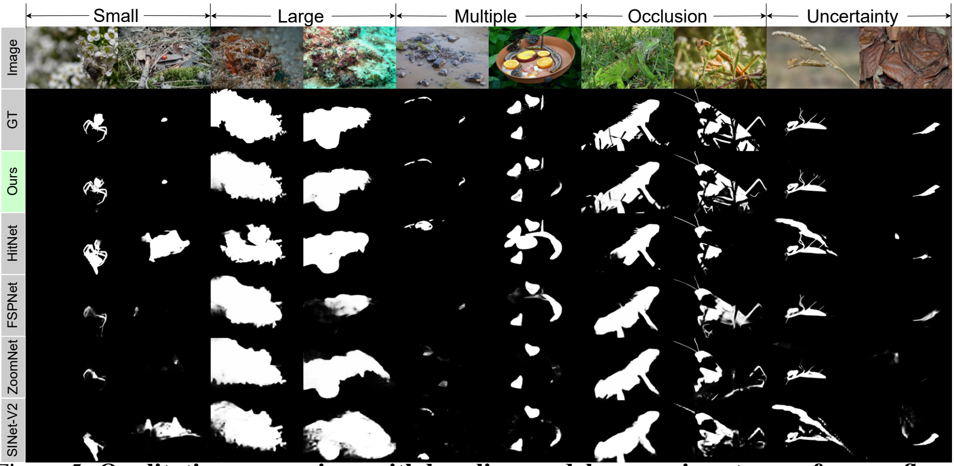

Figure 5: **Qualitative comparison with baseline models on various types of camouflaged objects.** Our model effectively captures diverse ranges of camouflages in the datasets.

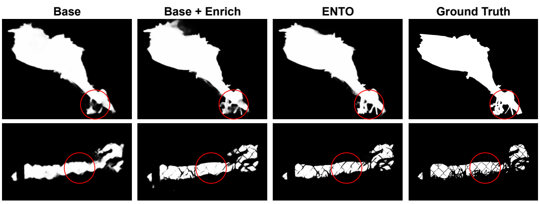

Figure 6: **Segmentation maps with ENTO components.** Enrich Decoder adds missing regions using COD-specific features, and Retouch Decoder finetunes the details near the edges.

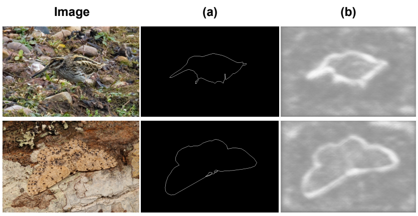

Figure 7: **Illustration of Spatial Attention by Retouch Decoder.** (a) ground truth of the edge, (b) spatial attention by our SAB.

Enrich Decoder recovers the details missing in segmentation map produced by base decoder and adding the Retouch Decoder sharpens these details to correctly match the true map.

Fig. 7 illustrates the visual representation of spatial attention in our Retouch Decoder. Notably, the attention scores mainly concentrate on the object boundaries, showing its predominant focus in refining these regions. In the final phase of decoding, Retouch Decoder effectively enhances intricate details regarding these edges and boundaries.

6 Summary

We present a novel decoding architecture ENTO to utilize high-resolution information extraction and address the complex boundary structures for camouflaged object detection. We propose Enrich Decoder and Retouch Decoder applicable to the base COD decoder, emphasizing feature channels beneficial for high-resolution segmentation by channel-wise attention and refining the segmentation map from the base decoder by spatial attention focusing on the edges and fine details, respectively. Our decoding architecture can be combined with different encoder backbones and shows superior results with Transformer backbones.

Acknowledgements

This work was supported by the New Faculty Startup Fund from Seoul National University, by Samsung Electronics Co., Ltd (IO230414-05943-01, RAJ0123ZZ-80SD), by Youlchon Foundation (Nongshim Corp.), and by National Research Foundation (NRF) grants (No. 2021H1D3A2A03038607/50%, RS-2024-00336576/10%, RS- 2023-00222663/5%) and Institute for Information & communication Technology Planning & evaluation (IITP) grants (No. RS-2024-00353131/25%, RS-2022-II220264/10%), funded by the government of Korea.

References

[1] Nagappa U Bhajantri and P Nagabhushan. Camouflage defect identification: a novel approach. In *Proc. of the International Conference on Information Technology (ICIT)*, 2006.

[2] Tianyou Chen, Jin Xiao, Xiaoguang Hu, Guofeng Zhang, and Shaojie Wang. Boundary-guided network for camouflaged object detection. *Knowledge-Based Systems*, 248:108901, 2022.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

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

[5] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *CVPR*, 2020.

[6] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. PraNet: Parallel reverse attention network for polyp segmentation. In *Proc. of the Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2020.

[7] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10): 6024–6042, 2021.

[8] Deng-Ping Fan, Ge-Peng Ji, Xuebin Qin, and Ming-Ming Cheng. Cognitive vision inspired object segmentation metric and loss function. *Scientia Sinica Informationis*, 6 (6), 2021.

[9] Seongsu Ha, Chaeyun Kim, Donghwa Kim, Junho Lee, Sangho Lee, and Joonseok Lee. Finding NeMo: Negative-mined mosaic augmentation for referring image segmentation. In *ECCV*, 2024.

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] Xiaobin Hu, Shuo Wang, Xuebin Qin, Hang Dai, Wenqi Ren, Donghao Luo, Ying Tai, and Ling Shao. High-resolution iterative feedback network for camouflaged object detection. In *AAAI*, 2023.

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

[14] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *CVPR*, 2022.

[15] Sookyung Kim, Hyojin Kim, Joonseok Lee, Sangwoong Yoon, Samira Ebrahimi Kahou, Karthik Kashinath, and Mr Prabhat. Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *WACV*, 2019.

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

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

[18] Maozhen Liu and Xiaoguang Di. Extraordinary MHNet: Military high-level camouflage object detection network and dataset. *Neurocomputing*, page 126466, 2023.

[19] Zhengyi Liu, Zhili Zhang, Yacheng Tan, and Wei Wu. Boosting camouflaged object detection with dual-task interactive transformer. In *Proc. of the International Conference on Pattern Recognition (ICPR)*, 2022.

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

[21] Yunqiu Lyu, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects. In *CVPR*, 2021.

[22] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *CVPR*, 2014.

[23] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *CVPR*, 2021.

[24] Sami Merilaita, Nicholas E Scott-Samuel, and Innes C Cuthill. How camouflage works. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1724): 20160341, 2017.

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

[26] Esther Parra-Mora, Alex Cazañas-Gordón, and Luís A da Silva Cruz. The effect of input size in deep learning semantic segmentation. In *Proc. of the IEEE Ecuador Technical Chapters Meeting (ETCM)*, 2022.

[27] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, 2012.

[28] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. BasNet: Boundary-aware salient object detection. In *CVPR*, 2019.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[30] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection. *arXiv:2105.12555*, 2021.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[32] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

[33] Jun Wei, Shuhui Wang, and Qingming Huang. F^3Net: fusion, feedback and focus for salient object detection. In *AAAI*, 2020.

[34] Haozhe Xing, Yan Wang, Xujun Wei, Hao Tang, Shuyong Gao, and Wenqiang Zhang. 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*, 2023.

[35] Yi Zeng, Pingping Zhang, Jianming Zhang, Zhe Lin, and Huchuan Lu. Towards high-resolution salient object detection. In *ICCV*, 2019.

[36] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *CVPR*, 2021.

[37] Miao Zhang, Shuang Xu, Yongri Piao, Dongxiang Shi, Shusen Lin, and Huchuan Lu. PreyNet: Preying on camouflaged objects. In *ACM MM*, 2022.

[38] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.

[39] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jing Qin. I can find you! boundary-guided separated attention network for camouflaged object detection. In *AAAI*, 2022.