MASNet: Multi-scale Aiming and Specification Network for Camouflaged Object Detection

Seungjai Bang[1] Joopyo Hong[2] Suyoung Kim[3] Seung Woo Ko[2]
[1]BDAI, Seoul National University [2]GSDS, Seoul National University
[3]MIPAL, Seoul National University

Abstract

*Camouflaged object detection (COD) is a challenging sub-task in object detection field with various application and has had significant performance improvements in recent years. Many COD models utilize ideas from traditional object detection and apply techniques that can better adapt the model to the camouflaged objects. However, as of now, there has not been an analysis of COD models on traditional object detection, which can pin-point to which features of the COD models actually are specific and useful to detecting camouflage. We show that scale factor is an important part of extracting features from image with camouflaged object and identify characteristics of COD models that can robustly expand to general object detection tasks. We finally propose a **MASNet** that couples COD-specific feature extraction method of using multi-scaled input with iterative specification module, while utilizing pyramidal transformer backbone. We show that our model outperforms most of the state-of-the-art models in COD benchmarks at many evaluation metrics, proving the effectiveness of multi-scaled input and transformer backbone to the success of COD models.*

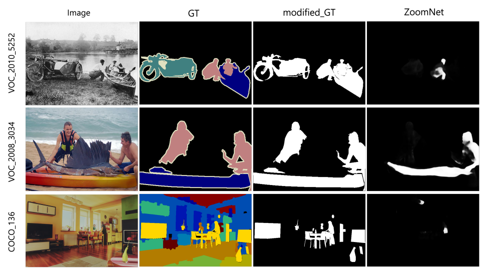
1. Introduction

Object detection is a widely researched area in computer vision, with several sub-tasks. Of these sub-tasks, camouflaged object detection (COD) [9] targets images that contain naturally or artificially camouflaged objects and aims to correctly segment them from the background. Success in COD tasks offer various ports of applications, such as Polyp [10] segmentation in medicine and surface defect detection [1].

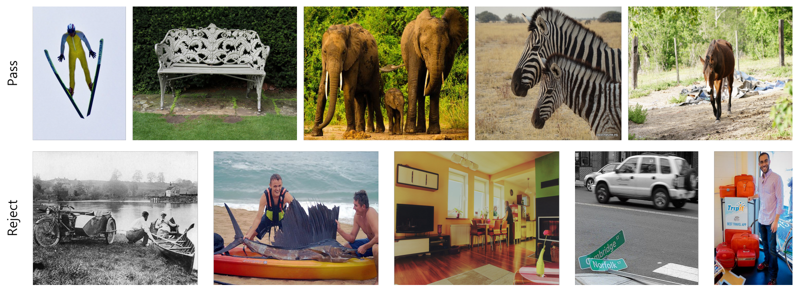
As camouflage by definition makes the object hard to distinguish from the background, developing a model that can identify and segment camouflaged object is a challenging task. Common features of these models is to utilize U-Net [25] like model architecture, with the encoder extracting features necessary for detecting camouflage and the decoder utilizing these features to produce a segmentation map. Most of the differences in these models lie in the feature extraction method. Current researches in using deep learning for COD utilize several techniques: incorporating auxiliary tasks, such as edge detection [3] and camouflage ranking [19], taking inspiration from biology, mimicking predators methods of hunting down camouflaged prey [7, 9], and applying recent advances in computer vision, such as vision transformer [14] and frequency regimes [31], to COD.

Although camouflaged object detection task can be considered as a specific case of general object detection (GOD) and segmentation task, there has not been a systematic analysis of COD models on GOD datasets. Such analysis can provide two important insights to COD task in general. First, by looking at the commonalities of COD models that have high performance increase in COD tasks compared to GOD tasks, we can properly assess what features of COD models are actually important for detecting and segmenting camouflage. In addition, by looking at COD models that are robust when applied to GOD, i.e. performance drop is low compared to other models, we can infer which parts of COD models can be expanded to the more general tasks. Using knowledge gained from such analysis, we suggest a COD model that can maximize the use of COD-specific features, while also being robust when applied to generalized object detection tasks. As we show later in Sec. 2, multi-scale features are needed to amplify COD-specific features, while coarse-to-fine decoding strategy is useful for generalized object detection.

In addition, recently, Transformer [27] has become the dominant architecture of choice in the computer vision researches for feature extraction. Although most of the COD models use ResNet [11], few COD models utilize Transformer backbone as their feature encoder. HitNet [13] and FSPNet [14] both utilize transformer backbone to extract features from camouflaged images and show very good performance on COD benchmarks. These recent advances sug-



(a)



(b)

Figure 1. **(a)** Examples of COD model failing at multi-class, multi-instance segmentation. **(b)** Examples of images passed and rejected during screening.

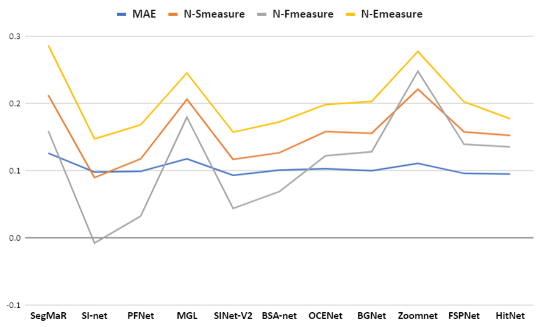


Figure 2. Performance differences between COD and GOD in SOTA COD models for 4 common COD evaluation metrics. The performance difference is normalized by model's performance to account for the true increase in performance.

gest that Transformer backbone can highly benefit the performance of COD models.

With the above mentioned findings, we suggest a new COD model architecture called **M**ulti-scale **A**iming and **S**pecification **Net**work (**MASNet**) that utilizes different scales of images to extract rich features from transformer backbone, with iterative specification of the target object. Our model surpasses previous state-of-the-art COD models under several evaluation criteria.

The main contributions of our project are summarized as follows:

- We analyze state-of-the-art COD models on GOD datasets and deduce commonalities of models that improves the most on COD tasks compared to GOD tasks and of models that show robust performance in GOD tasks.
- We propose a new COD model based on amplifying COD-specific features using multi-scale features, with its generality improved by iterative specification of the target object, which outperforms state-of-the-art models in this area in most of the evaluation criteria.
- We show that Transformer backbone structure helps to improve the performance of COD model.

2. Related Works

2.1. Camouflaged Object Detection.

Camouflaged object detection was first suggested as a sub-task to object detection by [9], providing a large-scale dataset (COD10K) to be used as baseline for all COD models. In response to this effort, numerous reseraches have tried to apply deep learning to detect and segment camouflaged objects.

Since camouflaged objects are hard to segment, some models utilize auxiliary tasks to help detect the objects. Rank-Net [19] utilizes a subsidiary task of camouflage object ranking, which aims to infer the level of camouflage in the objects. Additionally, since edge detection is key part of differentiating camouflaged object from the background, there are also efforts [3] to use edge detection as a auxiliary task to COD task.

SINet [9] and SINet-V2 [7] both take inspirations from biology, where the predator first searches for the prey in general sense and identifies the precise location. PFNet [21] also adopts a bio-inspired approach, where the positioning module (P) is used to identify the position of the camouflaged object and focus module (F) refines the fine details of it. In a similar manner, BSA-Net [32] mimics how humans discover camouflaged object using two-stream attention network.

Some COD models focus on the size of the objects in the image and introduce methods so that the models can extract features with different scales of an image. ZoomNet [22] achieves this through mixing features extracted from different scales. Instead, SegMaR [15] provides a model-agnostic framework, where images are iteratively magnified to better capture fine details of segmentation.

Recent efforts in COD aims to apply recent trends in computer vision to COD field. FSP-Net [14] utilizes vision transformers instead of ResNet backbone commonly used in COD models. Frequency-based model [31] was also introduced, utilizing high-frequency range of images to better detect the boundaries of the camouflaged objects. Another model [12] applies weak supervision to COD, as it is very costly to produce detailed annotation of camouflaged objects.

However, despite such efforts to expand the performance of COD models, there has not been a comprehensive study of what features of COD models are indeed helpful for detecting camouflage. Without such evaluation, performance increase in the COD models can just be attributed to the models utilizing better techniques for general object detection, rather than COD-specific methods. Thus, we believe that evaluation of COD models on general object detection is a necessary measure to identify accurately the characteristics of COD models that make them succeed in COD tasks.

2.2. Analysis of COD Models on Both COD and GOD Datasets.

In order to design an improved model architecture for camouflaged object detection task, we first analyze existing state-of-the-arts methods for COD task. We select 11 COD models pretrained with COD10K dataset and evaluate them on both COD datasets (COD10K [9], CAMO [16], CHAMELEON [26]) and traditional object detection datasets (Pascal-VOC [5], MS-COCO [17]). Incidentally, there is a critical limitation on the COD datasets in that they only contain single class per image. Therefore, models trained on the COD datasets show low performance on multi-class images, even if the models could fully capture the object feature information, as evidenced in Fig. 1a. For more accurate evaluation, we select images that only contain one identifiable instance or multiple identifiable instances of a single class. Few examples of such screening process are shown in Fig. 1b. As a result, we select 1075, 550 images from MS-COCO and Pascal-VOC, respectively and evaluate the 11 COD models on the screened datasets.

Multi-Scale Feature Input. We start with the hypothesis that the feature extraction method that a model should focus on are different in COD and GOD tasks. Fig. 2 shows the performance differences of the models on COD10K dataset and the COCO dataset for 4 common COD evaluation metrics. ZoomNet [22], MGL [30], and SegMaR [15] show big increases in performance because they perform much better on COD datasets than on GOD, implying that compared to other models, their feature extraction method is specific for COD task. The commonality of these methods is that they use methods to process different scales of the provided image, suggesting that image scale factor is an important feature to COD. Therefore, we design a model that can utilize such finding by providing multi-scale feature input to the model.

Coarse-to-Fine Iterative Decoder. Evidenced in Fig. 2, models robust to GOD are SINet [9], SINet-v2 [7], PFNet [21], and BSA-Net [32] that show the least performance drop on GOD dataset compared to COD dataset. We examine that the key model architecture all these models share is a module that explores low-level features again at the last phase and combines those features with the prediction map, basing their architecture on bio-inspired process. In detail, SINet and SINet-V2 both consist of search and identification module, and PFnet consists of positioning module and focus module. Both search and positioning module create coarse map with features extracted from backbone. Then, both identification and focus module combine the coarse map which contains high-level features with low-level features from backbone in order to create final fine-grained output map. BSA-net also is a coarse-to-fine model that initially creates coarse maps from attention blocks then refine maps with output of boundary guider module which extracts boundary information from low-level features. Most COD models' high-level features mainly contain COD-specific features while low-level features still contain general information feature. Therefore, combining low-level feature at the end would recover general image information, improving generality of model on different type of dataset. With this finding, we employ a coarse-to-fine iterative decoder to our model.

2.3. Transformer Backbone.

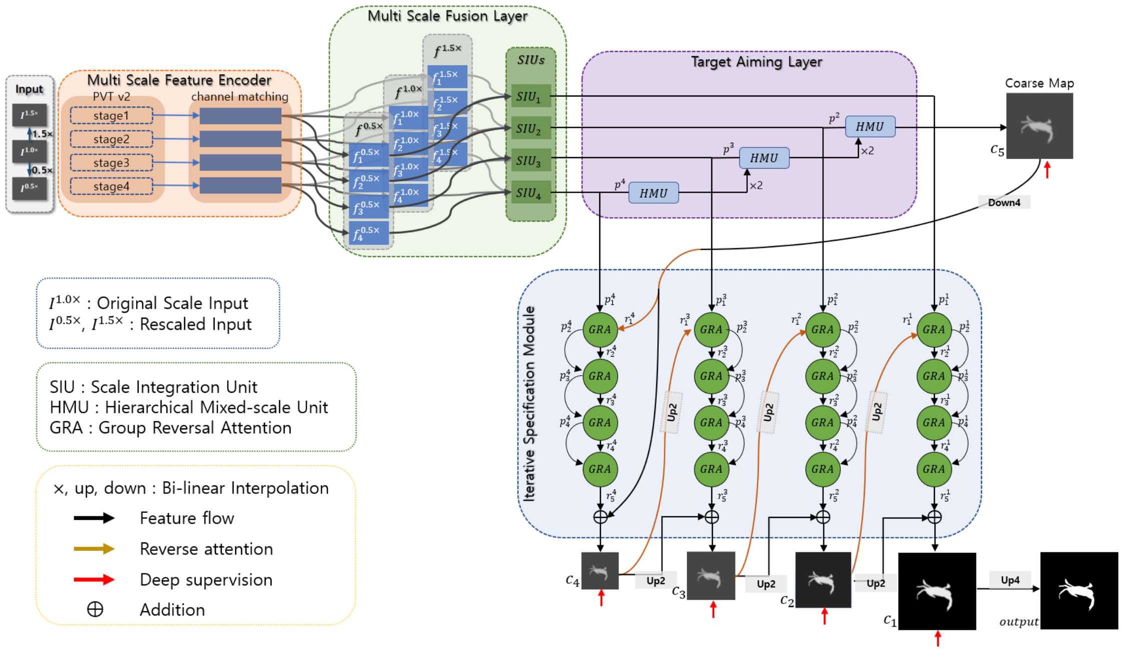
Recently, Transformer [27] is widely used in computer vison researches, due to its ability to extract useful features from images and videos. ViT [4] divides the images into patches, which are fed into the Transformer as tokens. In recent years, Pyramid Vision Transformer [28] was introduced, which improves usage of Transformers in images by provding a pyramid-style encoding structure that has versatility to be applied to many different computer vision tasks, such as object detction and segmentation.

Few studies in COD utilize Transformer backbone as their feature extractor. FSPNet [14] utilize vanilla ViT [4] as the backbone encoder for their model and show improved performance over previous COD models. HitNet [13] Pyramid Vision Transformer [28] to extract features for COD, also showing improved perforamance over previous models. These researches show that although the trend in COD was to use ResNet [11] models for feature extraction, utlizing Transformers can outperform models based on ResNet.

3. Methods

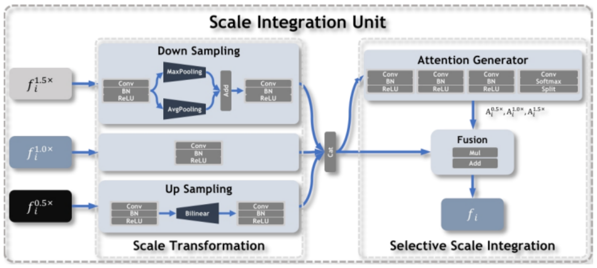
3.1. COD Problem Formulation

We simply formulate COD problem as follows: given an input of image containing a camouflaged object, $I_c \in \mathbb{R}^{H\times W\times 3}$, we want to produce an output segmentation map,

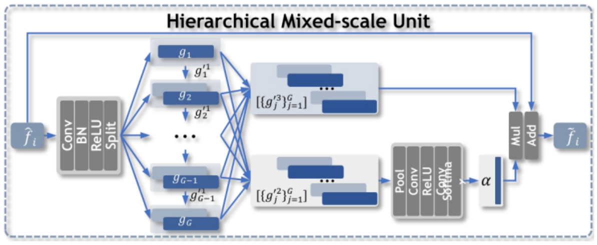


Input
Multi Scale Feature Encoder
PVT v2
channel matching
stage1
stage2
stage3
stage4
Multi Scale Fusion Layer
SIUs
Target Aiming Layer
HMU
Coarse Map
Down4
Iterative Specification Module
GRA
Up2
Up4
output
$I^{1.0\times}$: Original Scale Input
$I^{0.5\times}$, $I^{1.5\times}$: Rescaled Input
SIU : Scale Integration Unit
HMU : Hierarchical Mixed-scale Unit
GRA : Group Reversal Attention
×, up, down : Bi-linear Interpolation
Feature flow
Reverse attention
Deep supervision
Addition

Figure 3. Overall Architecture of MASNet.



(a)



(b)

Figure 4. Detailed archiecture of **(a)** SIU and **(b)** HMU [22].

$O_c \in \{0,1\}^{H\times W}$ that assigns value 1 for all pixels that belong to the camouflaged object and 0 for all other pixels.

3.2. Overall Architecture

Fig. 3 illustrates the overall architecture of MASNet, consisting of 3 main components: Multi-Scale Feature Encoder, Target Aiming Layer, and Iterative Specification Module. Initially, an input image is transformed into various scales and fed into the Feature Encoder in order to extract feature information at different scales. Encoder converts the input image pyramid using the PVTv2 transformer [28] into four feature maps and then matches the channel size of them into a 64-channel via a channel matching layer. Feature maps from the multi-scale inputs corresponding to the same feature level are combined using Scale Integration Unit (SIU) [22] of the Multi-scale Fusion Layer. The fused features are initially decoded using Hierarchical Mixed-scale Unit (HMU) [22], that constructs a coarse map at the end of the Target Aiming layer. Finally, the final prediction map is derived from the Iterative Specification Module which sequentially decode feature maps and a coarse map from the lower level to the higher level. The Iterative Specification Module produces and uses the guidance map on its own in every iterative step, while the coarse map from the Target Aiming layer is adopted for the guidance map in the first step. As a result, we obtain fine-grained segmentation maps by overcoming the situation where objects are vaguely hidden in their backgrounds.

3.3. Multi-scale Feature Extraction

As we mention in Sec. 2, extracting features from various scales of input images leads to performance increase specific to COD. Based on this finding, we use a strategy to process input images in multi-scales in the feature extraction process. We generate an image pyramid consisting of re-scaled images (0.5×, 1.0×, 1.5×) from the original input image using bilinear interpolation. Multi-scale input allows our model to find inconspicuous boundaries of camouflaged objects by utilizing information from detailed local context to global regional features. The feature extraction method of MASNet takes references from the ideas presented in [22].

Multi-scale Feature Encoder. We extract deep features for multi-scale input by using PVTv2 [28] as backbone which is pretrained on ImageNet-1K. We can extract four intermediate features from PVTv2 backbone, which form a feature pyramid similar to that extracted from CNN, making it easy to apply them in U-Net [25] like architectures. We use PVTv2-B2 which has 25.4M parameters comparable to ResNet50 [11] with 25.6M parameters. Channel matching modules adjust the channel size of each intermediate feature to the same size. Each channel matching layer is composed by five Conv-BN-ReLU layers which is inspired from ASPP [2] and modified for our architecture. Kernel sizes of channel matching modules are [1,3,3,3,1], dilation rates are [1,2,5,7,1], and each channel size is set to 64.

Multi-scale Fusion Layer. Since we use multi-scale images (0.5×, 1.0×, 1.5×) as input, the Multi-scale Feature Encoder also generates three differently scaled feature maps ($f_i^{0.5}$, $f_i^{1.0}$, $f_i^{1.5}$). The Multi-scale Fusion Layer uses the Scale Integration Unit(*SIU*) proposed in [22] to integrate the three feature maps into one unified feature map at each level, as shown in Fig. 4a. First, $f_i^{0.5}$ and $f_i^{1.5}$ are resized into the resolution of $f_i^{1.0}$ at each level. $f_i^{0.5}$ is scaled up via bilinear interpolation. And $f_i^{1.5}$ is down-sampled via hybrid pooling structure which consists of max pooling and average pooling layers. Conv-BN-ReLU layers are applied before and after resizing. The equally sized feature maps are then integrated with weights derived from Attention Generator. The Attention Generator receives the equally sized three feature maps consisting of respective 64-channels and derives attention maps corresponding to each size via series convolutional operation and softmax. These attention maps are used as weights in the integration task, so that feature maps are integrated into one while preserving crucial information from scales. We define SIU's operation as follows:

$$A_i = softmax(\Psi([U(f_i^{0.5}), f_i^{1.0}, D(f_i^{1.5})], \phi)) \quad (1)$$

$$p^i = A_i^{0.5} \cdot U(f_i^{0.5}) + A_i^{1.0} \cdot f_i^{1.0} + A_i^{1.5} \cdot U(f_i^{1.5}) \quad (2)$$

The Attention Maps are constructed via Eq. (1), where $\Psi(*, \phi)$ is the Attention Generator with Conv-BN-ReLU layer. Using these attention maps as weights, the integrated feature maps are obtained through Eq. (2) at each scale. Through this approach, it is possible to obtain COD specific feature maps, encapsulating distinctive information for each scale, which can be utilized in the decoding process.

3.4. Target Aiming Layer

The Target Aiming Layer is used to create the first coarse map of the Iterative Specification Module discussed in Sec. 3.5. The framework takes inspiration from the methodology of the decoder module of [22]. The Target Aiming Layer takes hierarchical feature maps as input and iteratively generates hidden decoded feature, ultimately achieving the same resolution with $f_2^{1.0}$. At each hierarchical iterative step, we utilize the Hierarchical Mixed-scale Unit (HMU) [22] to generate the coarse maps. HMU$_i$ at *i-th* feature level takes the input $\hat{p}^i$ composed as formulated as below Eq. (3). However, in for HMU$_4$, only the output p^4 from SIU$_4$ is utilized as the input.

$$\begin{aligned} \hat{p}^i &= p^i + Upsample(\tilde{p}^{i+1}) \quad (i = 2, 3) \\ \hat{p}^4 &= p^4 \end{aligned} \quad (3)$$

where $\tilde{p}^{i+1}$ is the result of HMU$_{i+1}$, and $Upsample()$ is a function that performs a two-fold bilinear interpolation upsampling.

Hierarchical Mixed-scale Unit. In the HMU, the information held by each channel is explored individually, and through channel-wise modulation, a coarse map is generated. An architecture of unit is illustrated in Fig. 4b. First, Input feature $\hat{p}^i$ undergoes 1×1 Conv-BN-ReLU layer and then is divided into G groups $\{p_j\}_{j=1}^{G}$ along the channels. Then, each group of features are further divided into three separate feature maps $\{p'^k_j\}_{k=1}^{3}$. However, before each group is divided, p_j is combined with the first feature map p'^1_{j-1} split from the preceding group. To address the problem of some groups lacking a previous group, we resolve it by performing the division without concatenation as an exception. The second feature maps of all groups $\{p'^2_j\}_{j=1}^{G}$ are concatenated and passed through a convolutional layer, as depicted in the Fig. 4b, to create the feature modulation vector(α). This group-specific feature modulation vector is utilized as weights to linearly integrate the third feature maps $\{p'^3_j\}_{j=1}^{G}$. Following the application of normalization and an activation function, the $\hat{p}^i$ is added to result in the final step. It can be summarized as follows:

$$\tilde{p}^i = \mathcal{A}(\hat{p}^i + \mathcal{CN}(\alpha \cdot [\{p'^3_j\}_{j=1}^{G}])) \quad (4)$$

where $\mathcal{A}()$, $\mathcal{CN}()$ denote the activation layer, Conv + Normalization layer, respectively.

In the final step, a sigmoid function is applied to the obtained $\hat{p}^2$ from the HMU$_2$ results to derive the Coarse map with 1/8 resolution of the original input image. Consequently, this coarse map effectively encompasses both the scale-wise and channel-wise information, and it is utilized in the first iteration of the Iterative Specification Module.

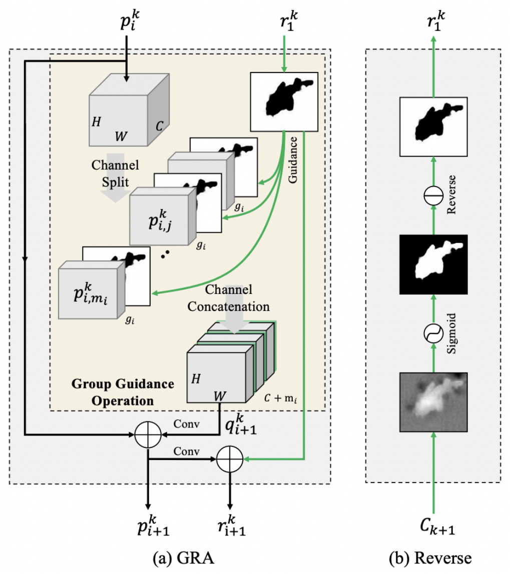


Figure 5. Details on **(a)** GRA Block and **(b)** Reverse Guidance [7]

3.5. Iterative Specification Module

As our coarse map (C_5) only captures rough spatial information of target object, Iterative Specification Module further explores the details of image features and specify the target object in an iterative manner. As depicted in Fig. 3, Iterative Specification Module consists of 4 iterative stages and each stage includes 4 GRA blocks which is proposed in [7] for multi-stage refinement process. More details about the GRA blocks and iterative stages will be addressed below.

Group-Reversal Attention (GRA). GRA block is a residual learning process utilizing reverse guidance and group guidance operation. Reverse guidance, shown in Fig. 5 (b) is a strategy to figure out discriminative concealed regions by erasing objects via sigmoid and reverse operation. [10]. The obtained output reverse attention guidance prior (r_1^k) can be formulated as:

$$r_1^k = \begin{cases} \ominus[\sigma(\delta_\downarrow^4(C_{k+1})), \mathbf{E}], k = 4, \\ \ominus[\sigma(\delta_\uparrow^2(C_{k+1})), \mathbf{E}], k \in \{1, 2, 3\}, \end{cases} \tag{5}$$

where $\delta_\downarrow^4$ and $\delta_\uparrow^2$ denote a x4 down-sampling and x2 up-sampling operation and σ is the sigmoid function. $\ominus$ is a reverse operation subtracting the input from matrix $\mathbf{E}$, in which all the elements are 1.

Group Guidance Operation (GGO) is a group-wise operation to utilize the reverse guidance prior (r_1^k) more effectively. [7] It consists of 2 steps: splitting candidate features (p_i^k) into multiple ($m_i = C/g_i$) groups along the channel-wise dimension, and periodically interpolating the guidance prior (r_1^k) among the split features ($p_{i,j}^k$). (Eq. (6))

$$\begin{aligned} &\textit{Step I: } \{p_{i,1}^k, ..., p_{i,j}^k, ..., p_{i,m_i}^k\} \leftarrow \mathbf{F}^S(p_i^k) \\ &\textit{Step II: } q_{i+1}^k \leftarrow \mathbf{F}^C(\{p_{i,1}^k, r_1^k\}, ..., \{p_{i,j}^k, r_1^k\}, ..., \{p_{i,m_i}^k, r_1^k\}), \end{aligned} \tag{6}$$

where $i \in \{1, 2, 3, 4\}, j \in \{1, ..., m_i\}, k \in \{1, 2, 3, 4\}$, and $g_i \in \{64, 8, 4, 2\}, \mathbf{F}^S, \mathbf{F}^C$ denotes the group size, channel-wise split function, and concatenation function. Thus, the GGO operation can be expressed as $\mathbf{F}^{GGO} : p_i^k \in \mathbb{R}^{H/2^k \times W/2^k} \rightarrow q_{i+1}^k \in \mathbb{R}^{H/2^k \times W/2^k \times (C+m_i)}$.

In each GRA block ($G_i^k, i \in \{1, 2, 3, 4\}, k \in \{1, 2, 3, 4\}$), candidate features ($p_i^k$ and r_1^k) are combined by GGO operation and residual connections are added to produce refined features (p_{i+1}^k and r_{i+1}^k) for the next GRA block. The formulas are as follow.

$$\begin{aligned} p_{i+1}^k &= p_i^k + g[\mathbf{F}^{GGO}[p_i^k, r_1^k; m_i]; \mathbf{W}_{GRA}^v], \\ r_{i+1}^k &= r_1^k + g[p_{i+1}^k; \mathbf{W}_{GRA}^w], \end{aligned} \tag{7}$$

where W_{GRA}^v denotes the convolutional layer with a 3×3 kernel for reducing the channel number from $C + m_i$ to C and W_{GRA}^w is also a 3×3 convolutional layer reducing the channel number to 1.

Iterative Specification Stages.

As shown in Fig. 3, in the first stage (k=1), coarse map (C_5) and feature extracted from SIU$_4$ are fed into the first GRA block. For stages $k \in \{2, 3, 4\}$, upsampled output of previous stage (C_{k-1}) and feature extracted from SIU$_{5-k}$ are fed into the first GRA block G_1^k , sequentially improving guidance for the next stage. The output of each stage is obtained with residual prediction, formulated as:

$$C_k = r_{i+1}^k + \delta(C_{k+1}), \tag{8}$$

where $\delta(\cdot)$ is $\delta_\downarrow^4$ when k=4 and $\delta_\uparrow^2$ when k= $\{1, 2, 3\}$. The output of the last stage (C_1) is utilized as the final prediction map after applying x4 up-sampling.

3.6. Loss Function

As we iteratively refine our segmentation map through the iterative specificaiton module, we apply supervision to each of the output maps and on the coarse map generated from target aiming layer, totaling to 5 different positions of supervision. In each case, the output map is first upscaled to ground truth size, then we utilize two loss functions commonly used in object detection tasks to train our model. Binary cross entropy (BCE), defined in [24] is applied to each pixel, accounting for pixel-level supervision to the output map. Intersection-over-union (IOU) loss, defined in [29], accounts for the overall overlap of the output map to the ground truth map. In both cases, each pixel is given a weight value, with higher weight given to pixels that are harder to detect. The loss at each supervision is given by

Method	COD10K				CAMO				CHAMELEON			
	$S_\alpha \uparrow$	$F^w_\beta \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F^w_\beta \uparrow$	$E_\phi \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F^w_\beta \uparrow$	$E_\phi \uparrow$	$M \downarrow$
HitNet [13]	**0.869**	**0.804**	**0.937**	**0.023**	**0.843**	**0.805**	**0.907**	**0.056**	**0.921**	**0.905**	**0.974**	**0.018**
FSPNet [14]	**0.851**	**0.736**	**0.903**	**0.026**	**0.860**	**0.808**	**0.916**	**0.051**	**0.907**	**0.850**	0.943	**0.022**
ZoomNet [22]	0.838	0.729	0.893	0.029	0.820	0.752	0.883	0.066	0.902	0.845	**0.952**	**0.023**
BGNet [3]	0.831	0.722	**0.903**	0.033	0.812	0.749	0.877	0.073	0.901	**0.850**	0.939	0.028
OCENet [18]	0.827	0.707	0.884	0.032	0.801	0.723	0.865	0.080	0.897	0.833	0.936	0.026
BSA-Net [32]	0.818	0.699	0.894	0.034	0.794	0.717	0.866	0.079	0.895	0.841	**0.946**	0.027
SINet-V2 [7]	0.815	0.680	0.864	0.037	0.820	0.743	0.885	0.070	0.888	0.816	0.929	0.030
MGL [30]	0.811	0.654	0.850	0.037	0.772	0.664	0.849	0.089	0.892	0.802	0.921	0.032
PFNet [21]	0.800	0.660	0.868	0.040	0.782	0.696	0.855	0.085	0.882	0.810	0.942	0.033
SINet [9]	0.776	0.631	0.867	0.043	0.745	0.644	0.825	0.092	0.872	0.806	0.938	0.034
SegMaR [15]	0.753	0.568	0.828	0.060	0.726	0.590	0.831	0.116	0.791	0.658	0.887	0.076
Ours(MASNet)	**0.880**	**0.794**	**0.922**	**0.022**	**0.872**	**0.824**	**0.919**	**0.048**	**0.922**	**0.873**	0.944	**0.023**

Table 1. Comparisons of different methods on Camouflaged Object Detection datasets. The best three results are highlighted in **red**, **green**, and **blue**, respectively.

Eq. (9) and by combining the loss values, we define the total loss function as in Eq. (10).

$$\mathcal{L} = \mathcal{L}_{wBCE} + \mathcal{L}_{wIOU} \tag{9}$$

$$\mathcal{L}_{total} = \sum_{i=1}^{5} \mathcal{L}(C_i, G) \tag{10}$$

4. Experiments

4.1. Experiment Setup

Datasets. We selected three widely-used COD datasets: COD10K [9], CAMO [16], CAMELEON [26] and two traditional object detection datasets: MS-COCO [17], Pascal-VOC [5]. CAMO consists of 1,250 camouflaged and 1,250 non-camouflaged images. CHAMELEON contains 76 hand-annotated images. COD10K includes 5,066 camouflaged, 3,000 background and 1,934 non-camouflaged images. We train the model using only camouflaged images from COD10K and CAMO datasets and evaluate the model on CHAMELEON and test splits of COD10K and CAMO.

Evaluation Criteria. For two types of datasets, we both applied 4 common COD evaluation metrics: S-measure (S_α) [6], weighted F-measure (F^w_β) [20], E-measure (E_ϕ) [8], and mean absolute error (M) [23]. S-measure quantifies the structural similarity between the model output and the ground truth, which is important in COD tasks, which usually contain complex shapes of objects. Weighted f-measure is a modified version of F-measure that provide more reliable evaluation. 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.

Implementation Details. The framework of MASNet is implemented using PyTorch. A pre-trained PVTv2 [28] on ImageNet-1K is utilized in feature extraction. During the model training, a learning rate is set to 0.05 and a scheduler linearly increases the learning rate for the first half of training to the set value and decreases it for the other half. WE use SGD optimizer with 0.9 momentum and 0.0005 weight decay. A batch size of 16 is chosen, and the training is performed on 100 epochs. Our model was trained in a development environment using 4 NVIDIA TITAN V, and the training took approximately 6 hours in total.

4.2. Comparison to COD Benchmarks

Quantitative Comparison. The quantitative comparison results are shown in Tab. 1. Our method surpassed all the other existing methods on CAMO dataset on all 4 evaluation metrics. On the other 2 datasets: COD10K and CHAMELEON, our method was in the top-3 performance in all metrics except E-measure (E_ϕ) on CHAMELEON. These results demonstrate that proposed model has effectively extracted and captured the camouflaged features. Especially, our method showed large advantages on S-measure (S_α) taking the first place in every dataset, which indicates that proposed method is effective in capturing structural information.

Qualitative Comparison. The qualitative comparison results are shown in Fig. 6, comparing our MASNet and other recent models including SINet [9], SINetV2 [7], ZoomNet [22], and HitNet [13] on COD10K dataset. Our model tends to capture well both overall structure, while also preserving the fine details, such as slim leg parts, compared to other models.

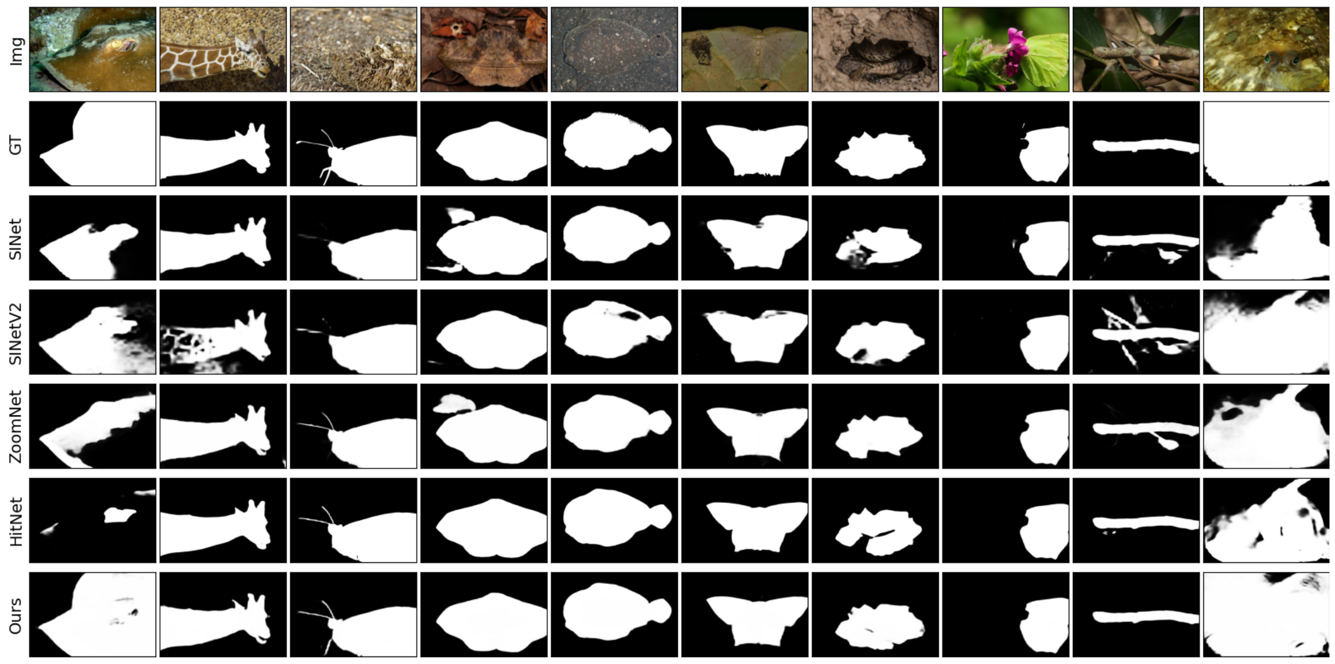


Figure 6. Visual comparisons of some recent COD methods and ours on different types of samples. Please zoom in for more details.

Table 2. **Ablation study on different components of MASNet.** Each row denotes changes from the full model. Performances on COD10k test dataset is shown.

Methods	$S_\alpha \uparrow$	$F^w_\beta \uparrow$	$E_\phi \uparrow$	$M \downarrow$
MASNet	**0.880**	**0.794**	**0.930**	**0.022**
$\rightarrow$ 3 specification iterations	0.879	0.793	0.928	0.022
$\rightarrow$ 3 stage GRA	0.877	0.788	0.926	0.023
$\rightarrow \mathcal{L} = \mathcal{L}_{wBCE}$ (no $\mathcal{L}_{wIOU}$)	0.871	0.747	0.900	0.026
$\rightarrow$ Original Input Scale	0.853	0.744	0.897	0.028
$\rightarrow$ ResNet Backbone	0.833	0.713	0.884	0.033

4.3. Ablation Studies

Effectiveness of Transformer backbone. As shown in Tab. 2 replacing the Tranformer PVT v2 [28] with ResNet [11] significantly reduces the performance of the model on all four evaluation metrics. Since PVTv2-B2 has comparable parameter size with ResNet50, we can conclude that Transformer architecture itself is useful in extracting useful features for COD.

Effectiveness of Multi-scale Input. To test our hypothesis that multi-scaled input is essential for success camouflaged object detection, we feed the model with three parallelized original scale (1.0) inputs. The model performance drops on all metrics, showing that multi-scale input is necessary to the model success.

Effectiveness of IoU Loss. As described in Sec. 3.6 IOU loss is responsible for overall overlap between the output map and the ground truth. By removing this loss component, we see that the overall performance decreases, but especially weighted F-measure

Effectiveness of GRA Module. In our model, we apply 4 stages of GRAs with 4 iterations to provide guidance from the generated coarse maps. Reducing the stages to just 3 or iterations to 3 both results in slight performance drop.

5. Conclusion

Analyzing performance of COD models on general object detection tasks reveals important characteristics of COD models. Scale-factor is an important part of model feature that improves the model's ability to specify on camouflaged objects, while recombination of low-level features using coarse-to-fine model helps to generalize the model to general object detection. Our model, **MASNet**, incorporates these findings into a single model, with powerful Transformer backbone as our feature encoder, showing outstanding performance in all COD benchmarks.

With these improvements, we believe that further amplifying multi-scale features using Transformer architecture, such as through cross-attention between different scales, may be an interesting following work. In addition, minor issues with our model, such as loss of information in the upscaling of the coarse maps and the feature level at which the coarse map is generated from, should be analyzed for further improvement.

References

[1] Nagappa U Bhajantri and P Nagabhushan. Camouflage defect identification: a novel approach. In *9th International Conference on Information Technology (ICIT'06)*, pages 145–148. IEEE, 2006. 1

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 5

[3] Tianyou Chen, Jin Xiao, Xiaoguang Hu, Guofeng Zhang, and Shaojie Wang. Boundary-guided network for camouflaged object detection. *Knowledge-Based Systems*, 248:108901, 2022. 1, 2, 7

[4] 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. *arXiv preprint arXiv:2010.11929*, 2020. 3

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 3, 7

[6] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017. 7

[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. 1, 2, 3, 6, 7

[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. 7

[9] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2777–2787, 2020. 1, 2, 3, 7

[10] 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 *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23*, pages 263–273. Springer, 2020. 1, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3, 5, 8

[12] Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson WH Lau. Weakly-supervised camouflaged object detection with scribble annotations. *arXiv preprint arXiv:2207.14083*, 2022. 3

[13] Xiaobin Hu, Deng-Ping Fan, Xuebin Qin, Hang Dai, Wenqi Ren, Ying Tai, Chengjie Wang, and Ling Shao. High-resolution iterative feedback network for camouflaged object detection. *arXiv preprint arXiv:2203.11624*, 2022. 1, 3, 7

[14] 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. *arXiv preprint arXiv:2303.14816*, 2023. 1, 2, 3, 7

[15] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4713–4722, 2022. 2, 3, 7

[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. 3, 7

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3, 7

[18] Jiawei Liu, Jing Zhang, and Nick Barnes. Modeling aleatoric uncertainty for camouflaged object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1445–1454, 2022. 7

[19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11591–11601, 2021. 1, 2

[20] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 248–255, 2014. 7

[21] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8772–8781, 2021. 2, 3, 7

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2160–2170, 2022. 2, 3, 4, 5, 7

[23] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE conference on computer vision and pattern recognition*, pages 733–740. IEEE, 2012. 7

[24] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7479–7489, 2019. 6

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmen-

tation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1, 5

[26] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. *Unpublished manuscript*, 2(6):7, 2018. 3, 7

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3

[28] 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. 3, 4, 5, 7, 8

[29] Jun Wei, Shuhui Wang, and Qingming Huang. F^{3}net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12321–12328, 2020. 6

[30] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12997–13007, 2021. 3, 7

[31] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4504–4513, 2022. 1, 3

[32] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3608–3616, 2022. 2, 3, 7