SdalsNet: Self-Distilled Attention Localization and Shift Network for Unsupervised Camouflaged Object Detection

Peiyao Shou[1], Yixiu Liu[1]*, Wei Wang[2], Yaoqi Sun[1], Zhigao Zheng[3], Shangdong Zhu[1], Chenggang Yan[1]

[1]Hangzhou Dianzi University, China
[2]Shenzhen Campus of Sun Yat-sen University, China
[3]Wuhan University, China
peiyaoshou@gmail.com, {liuyixiu,syq,zhushd,cgyan}@hdu.edu.cn,
wangwei29@mail.sysu.edu.cn, zhengzhigao@whu.edu.cn

Abstract

Unsupervised camouflaged object detection (UCOD) poses significant challenges, primarily attributed to the absence of human labels. Existing UCOD methodologies, leveraging attention mechanisms, often struggle to achieve precise localization of camouflaged objects. To overcome this limitation, we introduce a groundbreaking fully unsupervised algorithm for attention-guided camouflaged object localization, shift, and inference, termed the self-distilled attention localization and shift network (SdalsNet). In this study, we formulate an attention localization methodology aimed at accurately identifying the central coordinate of the camouflaged object. Furthermore, we propose four distinct loss functions tailored to refine the precision of attentional positioning. These loss functions effectively constrain the distances between three types of class tokens, facilitating seamless attentional shifting across the input sample. Additionally, we design a sophisticated prediction inference technique to reconstruct the binary output of an attention map, thereby providing a comprehensive understanding of the detected camouflaged objects. Experimental results on four challenging COD benchmark datasets corroborate the effectiveness of our proposed approach, demonstrating notable superiority over state-of-the-art methods.

Code — https://github.com/Alpha-Orionis/SdalsNet.

Introduction

In real-world scenarios, there exists a specific domain, camouflaged object detection (Fan et al. 2020), which aims to segment camouflaged objects that blend into the background. Most existing COD methods are based on the supervised ImageNet (Deng et al. 2009) pre-trained backbone [(He et al. 2016), (Dosovitskiy et al. 2021), (Liu et al. 2021), (Tan and Le 2019)] and the encoder-decoder framework [(Zhang et al. 2023), (Ji et al. 2023), (Zhu et al. 2022), (Hu et al. 2023)] trained on the well-labeled COD dataset (Fan et al. 2022). Due to the small amount of data in existing COD datasets and the challenging COD task, these models lack generalization ability when facing realistic scenarios. In order to apply the model to the real world, it is important to investigate unsupervised COD models that do not use any human annotations throughout the training process.

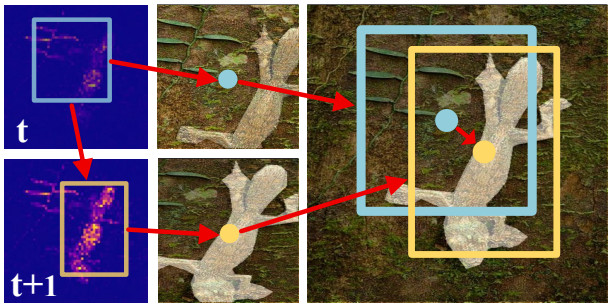


Figure 1: The t and $t+1$ iterations of the attention localization and attention shifting process in SdalsNet.

Most existing UCOD models are end-to-end approaches [(Melas-Kyriazi et al. 2022), (Zhou et al. 2023), (Siméoni et al. 2023), (Zhang and Wu 2023)] that attempt to obtain pixel-accurate segmentation models of camouflage objects. However, the existing pre-trained models are based on classification tasks such as ImageNet, and most of the targets in the pre-training data set are clearly distinguished from the background, which has a large gap with the COD datasets. Therefore, it is difficult to obtain pixel-wise accurate segmentation results using these models.

Inspired by FOUND (Siméoni et al. 2023) and UCOD-DA (Zhang and Wu 2023), ViT models pre-trained (Dosovitskiy et al. 2021) on ImageNet can produce good attention responses for COD tasks. However, these models are end-to-end and the network outputs pixel-wise segmentation results. Both FOUND and UCOD-DA freeze Vits model and train a 1×1 convolutional network to output pixel-wise binary segmentation maps. Since the training of transformer models requires a large amount of accurately labeled data, it is difficult to be used for unsupervised end-to-end COD tasks. In this paper, we propose a non-end-to-end UCOD model for the first time, which directly derives the prediction map from the attention map. The network takes the points with high attention as the target and sets them 1, and the

*Yixiu Liu is the corresponding author.

points with low attention as the background and sets them 0, so as to realize the segmentation of the target. In order for the network to generate better attention maps to optimize the segmentation results, we design a self-supervised training process to train and optimize the network attention in an unsupervised manner. Different from the end-to-end model, the model trains a simple classification task as a proxy task instead of a complex pixel segmentation task. The model only needs to focus on the overall features of the object, and does not have to perform pixel-by-pixel segmentation. DINO (Caron et al. 2021) confirmed that simple classification tasks can be used to train ViT models.

As shown in Figure 1, the training of the network is an iterative optimization process. It mainly contains two parts: attention localization and shift. In t iteration, the teacher model locates the target through the attention map and selects the corresponding small picture from the original picture box, and the student model learns the characteristics of the target location through the small picture and optimizes the attention map. In the figure, the blue and yellow boxes represent the target localization and the corresponding box selection pictures for the two iterations, respectively. The variation of the blue and yellow boxes demonstrates the iterative optimization process of the network. At $t+1$ iterations, the center of attention localization is shifted. From this, the network iteratively optimizes the attention map step by step.

The training process of the network is divided into two processes: attention localization and attention transfer. The inference process is divided into attention map normalization and denoising. Attention localization calculates the attention center point through the distance matrix weighted attention map, and uses it to locate the target position. The distance matrix contains the geometric distances between points in all attention maps. After weighting, points that are far away from the attention center region will calculate a higher sum value since they are all far away from the high attention points. Conversely, the points located near the center are all close to the high attention point, and the sum value is low. By finding the lowest weighted sum value, the attention center point can be obtained. The attention shift will pull the foreground small image and full image results closer, push the background small image and full image results further, and push the foreground and background small image results further away. This makes the network focus on the target region, output the features of the target region and ignore the background region, and move the attention center to a better location. Normalization can transform the continuous attention map into a discrete binary map, and denoising can remove the external noise and internal noise in the binary map to obtain better segmentation results.

Our main contributions are as follows:

- We propose to think about the object discovery problem in another way, by finding the regions in the picture where the network's attention is focused, rather than looking for the object directly.
- We design a first non-end-to-end UCOD model that takes the simpler classification task as a proxy task and directly utilizes the attention map to obtain the segmentation results.
- Extensive experiments on four challenging COD benchmarks show that our SdalsNet achieves state-of-the-art (SOTA) performance without using any human labels.

Related Work

UCOD Methods

There are some UCOD methods that have achieved good results in recent years. A2S-V2 (Zhou et al. 2023) mines knowledge from noisy labels generated by the pre-trained network and generates high-quality pseudo-labels to train the model. This network will directly output pixel-wise segmentation results. SelfMask (Shin, Albanie, and Xie 2023) performs k-means clustering on network features to obtain an object mask, which is used to train an object detection and segmentation model to solve the unsupervised object detection task. TokenCut (Wang et al. 2022) uses the normalized graph cut technique to segment the foreground object, treats the network feature as a graph and performs graph cuts, and groups similar regions and outputs as foreground objects. SpectralSeg (Melas-Kyriazi et al. 2022), which uses graph partitioning method to extract image features for self-supervised learning.

Self-Distillation Learning

In recent years, self-distillation [(Zagoruyko and Komodakis 2017), (Hou et al. 2019)] methods have been applied to unsupervised object detection and segmentation, and have achieved very good results. For example, FOUND (Siméoni et al. 2023) uses a pre-trained model of the self-distillation framework DINO (Caron et al. 2021), which is trained with DINO as a provider of pseudo-labels. UCOD-DA (Zhang and Wu 2023) similarly uses the pseudo-labels provided by the DINO framework and uses the DINO pre-trained model as the encoder. Similarly to the above approach, we also used the DINO pre-trained model. However, our model does not freeze, ViT parameters are updated during training to optimize model attention.

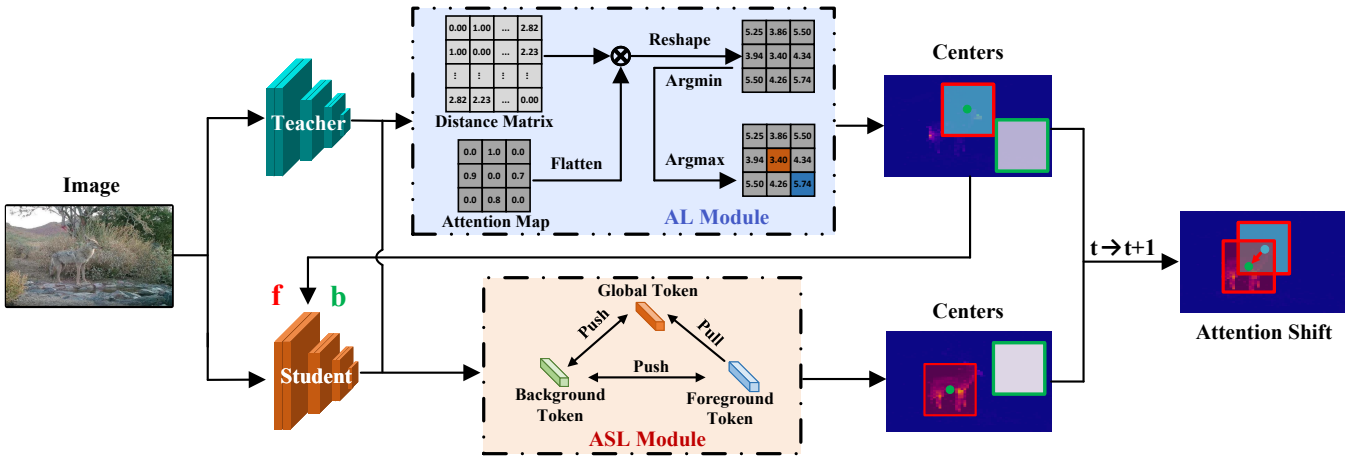
Methodology

Network and Training Strategy

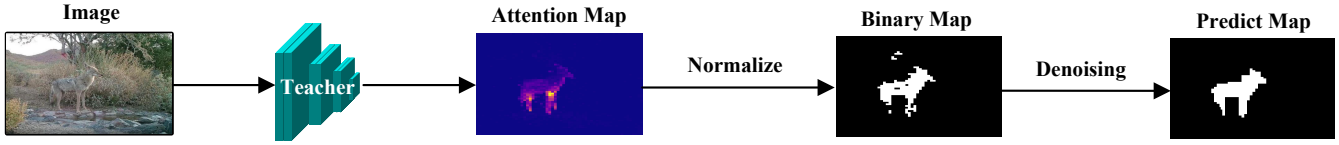
SdalsNet adopts the Vision Transformer (ViT) as the encoder Backbone architecture, and uses a class-head to map the vectors output by the vit to a high-dimensional space. The network is divided into a teacher model and a student model. The two models have the same architecture, which is composed of a vit and a class-head. Training using self-distillation. In the self-distillation training process, the teacher model does not calculate the gradient, and the student model calculates the gradient and does backpropagation to update the parameters. After the student model parameters are updated, the teacher model parameters are updated using the formula:

$$T_{new} = \lambda \times S_{new} + (1 - \lambda) \times T_{old}, \tag{1}$$

where T_{old} represents the parameters before the teacher model is updated, T_{new} represents the parameters after the



(a) Schematic diagram of architecture and training process of SdalsNet



(b) Schematic diagram of inferencing process of SdalsNet

Figure 2: The AL module shows an example of locating the 3×3 attention map output by the teacher model. $\otimes$ means matrix cross product. The foreground position obtained after localization is represented by the red box, and the green color represents the background center. In ASL module, "pull" means that two tokens should be closer together, "push" means that they should be further apart, the Global Token is the teacher model output and the other two tokens are the student model output.

teacher model is updated, and S_{new} represents the parameters after the student model is updated. The value of λ comes from the update method of the teacher model in the paper: DINO(Caron et al. 2021).

The model training process is divided into two stages. In the first stage, we train the class-head to adapt the task of our UCOD, at this time, only the class-head is trained and vit parameters are loaded into DINO's pre-trained model and frozen. In the second phase, we alternately train our ViT and class-head, which alternately freeze/unfreeze and train. By alternating training one part at a time, vit and class-head can better adapt to each other, enhancing the robustness of the network and reducing the impact of overfitting.

More specifically, the training process is divided into two parts: attention localization and attention shifting. In the attention localization process, vit of the teacher model generates the attention map, and the network locates objects from the attention map through the attention localization module(ALM). In the attention shift learning(ASL), the student model will learn the information of the area where the object is located, so that the attention is focused on the area where the object is located. The above two parts are shown in Figure 2 (a).

In the inference process of the model, the binary attention map is used. The points with high attention are regarded as target pixels and set to 1, and the background is set to 0, so as to binarize the attention map, and then obtain the final UCOD binary segmentation result after denoising. The above process are shown in Figure 2 (b).

Attention Localization Module

In order to localize the object from the attention map, we need to get the center of the high attention region, which is the center of the region where the object is located. First, the network computes the geometric distance of each element of the attention map to every other element. The result is a matrix consisting of $L \times L$, the distance matrix, where L is the number of elements in the attention map. The distance matrix D is computed as follows:

$$D_{i,j} = \sqrt{(i//w - j//w)^2 + (i\%w - j\%w)^2}, \quad (2)$$

where $i, j \in (0, h \times w - 1)$, and $D_{i,j}$ denotes the geometric distance from element i to element j in attention map, and . The shape of D is $L \times L$, where $L = H \times W$, which denotes the number of elements in the attention map.

A distance-weighted attention map can be obtained by D-weighting and summing the attention maps:

$$A^*(j) = (\sum_{i=0}^{h \times w - 1} D_{i,j} \times A_i) \,|\, j \in (0, h \times w - 1), \quad (3)$$

Here, A represents the attention map of the last layer in the ViT, A_i denote the i-th element in the attention map, and

$A^*(j)$represents the distance from the corresponding position $A(j)$to other elements in the attention map multiplied by the values of other elements and then summed.

The position corresponding to the minimum of A^* is the center point weighted by the attention distance, which is the attention geometric center of the foreground C_f. We use $\arg\min$ to get the Geometric center coordinates:

$$C_f(c_h, c_w) = \begin{cases} c_h = \arg\min\limits_j \ A^*(j) \ //w \\ c_w = \arg\min\limits_j \ A^*(j) \ \%w \end{cases}. \tag{4}$$

The background center of attention C_b is inversely maximized, calculated using $\arg\max$:

$$C_b(c_h, c_w) = \begin{cases} c_h = \arg\max\limits_j \ A^*(j) \ //w \\ c_w = \arg\max\limits_j \ A^*(j) \ \%w \end{cases}. \tag{5}$$

Finally, we box the images around the coordinates as the training data. The original image I input is represented as

$$I = \begin{pmatrix} I_{0,0} & I_{0,1} & \cdots & I_{0,w-1} \\ I_{1,0} & I_{1,1} & \cdots & I_{1,w-1} \\ \vdots & \vdots & \ddots & \vdots \\ I_{h-1,0} & I_{h-1,1} & \cdots & I_{h-1,w-1} \end{pmatrix}, \tag{6}$$

where $I_{x,y}$ denotes the data corresponding to the position of the image, the color picture should be three-channel. Intercept the box with size l around the center of attention as the data for subsequent training:

$$I^* = \begin{pmatrix} I_{c_h^*,c_w^*} & I_{c_h^*,c_w^*+1} & \cdots & I_{c_h^*,c_w^*+2l} \\ I_{c_h^*+1,c_w^*} & I_{c_h^*+1,c_w^*+1} & \cdots & I_{c_h^*+1,c_w^*+2l} \\ \vdots & \vdots & \ddots & \vdots \\ I_{c_h^*+2l,c_w^*} & I_{c_h^*+2l,c_w^*+1} & \cdots & I_{c_h^*+2l,c_w^*+2l} \end{pmatrix}, \tag{7}$$

where $c_h^* = c_h - l$ and $c_w^* = c_w - l$. Each image I results in two I^*, corresponding to the image around the foreground attention center and the image around the background attention center.

Attention Shift Learning

There are three kinds of images that are input to the student network, which are the global(full-size) image, the image around the foreground attention center, and the image around the background attention center.

The first is the self-distillation loss of the model, the output of the student model should be close to the teacher model, so the feature vector corresponding to the global image output by the student model should be the same as the output of the teacher model. The following loss function serves this purpose function:

$$\mathcal{L}_g = \sum_{i=0}^{d} |T_i - S_i^g| \, /d, \tag{8}$$

where T_i represents the i-th element of teacher model output, S_i^g represents the i-th element of the student model output corresponding to the global image, and d is the length of the feature vector.

Next are three loss functions that shifts attention. The output feature vector corresponding to the image of the foreground attention region should also be similar to the output of the teacher model, and the loss is calculated in the same function:

$$\mathcal{L}_f = \sum_{i=0}^{d} \left| T_i - S_i^f \right| /d, \tag{9}$$

where S_i^f represents the i-th element of the student model output corresponding to the image around the foreground attention center.

With the above two loss functions, student network can achieve a pull operation between the features of the whole image and the foreground features, which will enable the network to better extract the features of the high attention regions of the image.

Furthermore, the student model output should be far away from the background information, and when the student model input is the background region image, its output should be far away from the teacher model output. The following loss function can push away the teacher model output and the corresponding student model output of the background region image:

$$\mathcal{L}_b = 1 - \left(\sum_{i=0}^{d} \left| T_i - S_i^b \right| /d \right), \tag{10}$$

where S_i^b represents the i-th element of the student model output corresponding to the image around the background attention center.

At the same time, the image of the foreground region and the image of the background region of the student model itself are pushed away, which can strengthen the distinction between the foreground and background. Using a equation similar to $\mathcal{L}_b$, we obtain the following loss function:

$$\mathcal{L}_p = 1 - \left(\sum_{i=0}^{d} \left| S_i^f - S_i^b \right| /d \right). \tag{11}$$

Using the above two loss functions, the student network can inhibit the extraction of background features by the network, so that the network will pay attention to the foreground region instead of the background region.

The final total loss function is obtained by summing the four loss functions:

$$\mathcal{L}_{total} = \mathcal{L}_g + \alpha \mathcal{L}_f + \beta \mathcal{L}_b + \gamma \mathcal{L}_p. \tag{12}$$

In this equation, α, β, γ is the three hyper-parameters used to regulate the weights between the four loss functions.

Prediction Inference

After the model has a good attention to the foreground region, we use the attention map of the teacher model to extract the image segmentation result.

The attention map needs to be normalized to get the attention map with 0,1 distribution. Define the formula $f(x, y)$ as

follows:

$$f_i(x,y) = \begin{cases} 0 & x_i \leq y_i \\ 1 & x_i > y_i \end{cases}, \tag{13}$$

where x and y are two sequences of equal length, the output of function $f(x,y)$ is also a sequence of equal length, x_i and y_i are the elements of their respective sequences, and f_i is the element of the output sequence of the function, obtained from x_i and y_i.

The formula $f(x,y)$ is used to binarize the attention map. In the attention map, the elements greater than the mean value are set to 1, and the elements less than or equal to the mean value are set to 0. The following formula is used:

$$B_{ha} = f\left(A, Mean(A)\right), \tag{14}$$

were B_{ha} represents the high attention binary map, where 1 represents the high attention region(foreground region), and 0 represents the low attention region(background region). $Mean$ denotes the mean function, and $Mean(A)$ is the mean of the attention map A.

B_{ha} has a lot of noise. Here, the following two ways are used to remove these noise, which are the outer noise removal formula and the inner noise removal formula. Here is the outer noise removal formula:

$$B_{od} = f(AvgPool(B_{ha}),\ Mean(AvgPool(B_{ha}))), \tag{15}$$

where B_{od} represents the outer denoising binary map, and $AvgPool$ represents the Average-Pooling operation with a step size of 1, the pooling does not change the size of the binary map; here, the kernel-size of Average-Pooling is 33, and the padding is 16.

After this operation, the noise of the image will be zeroed out, but at the same time, the holes inside the image may be removed as noise. A inner noise removal formula that can correctly distinguish and remove the internal noise can solve this problem. Inner noise removal formula also uses Average-Pooling, and here is the formula:

$$B_{id} = f(AvgPool(B_{ha}),\ 0.5), \tag{16}$$

where B_{id} represents the inner denoising binary map, and the kernel-size of Average-Pooling is 5, padding is 2, and step is 1, which means that the output of this pooling does not change size of the binary map either. Because B_{id} is a 0-1 distributed binary graph, an intermediate value of 0.5 is chosen as the threshold.

Finally, taking the intersection of B_{od} and B_{id}, the final segmentation result can be obtained.

$$B_{pre} = B_{od} \cap B_{id}, \tag{17}$$

where B_{pre} represents the prediction binary map, this resulting map, is a binary map with 0-1 distribution, where 1 represents the pixel where the object is located and 0 represents the pixel where the background is located.

Experiments

Datasets and Evaluation Metrics

Datasets There are four popular datasets in the COD field: a) CAMO (Lea et al. 2019) has 1,250 camouflaged images.b) COD10K (Fan et al. 2022) is the largest COD training dataset with 3,040 training images.c) CHAMELEON (Skurowski et al. 2017) is about cryptic hidden animals masked in environment. d) NC4K (Lv et al. 2021) is the largest testing dataset, includes 4,121 samples. Following the protocol of (Fan et al. 2022), we train our model on the hybrid dataset (i.e., COD10K-Tr + CAMO-Tr) with 4,040 samples and evaluate our method on above four benchmarks (see Table 1).

Evaluation Metrics Following (Fan et al. 2022), we use four commonly used metrics for the evaluation: structure measure (S_α) (Fan et al. 2017), enhanced-alignment measure (E_ϕ^{max}) [(Deng Ping et al. 2018), (Deng Ping et al. 2021)], F-measure (F_β^{max}) [(Borji et al. 2015), (Zhuge et al. 2023)], and mean absolute error (M).

Implementation Details

The network in this paper is built using PyTorch and trained on a NVIDIA RTX 3090 GPU with a batch size of 12. We use the Adam optimizer to adjust the learning rate. The maximum number of training epochs is 9. To standardize the input, the images are resized to 224×224. We choose ViT-Small to build our network, where the patch-size is 8, embed-dim is 384, num-heads is 6 and has a total of 12 layers of transformer blocks. The class-head of the network is a fully connected neural network with three layers, the input dimension is 384, and the output dimension is 65536.

Comparison with the State-of-the-Arts

We compare the proposed SdalsNet with seven recent state-of-the-art unsupervised object segmentation model, including TokenCut(Wang et al. 2022), SelfMask(Shin, Albanie, and Xie 2023), SpectralSeg(Melas-Kyriazi et al. 2022), A2S-v2(Zhou et al. 2023), FOUND(Siméoni et al. 2023), UCOD-DA(Zhang and Wu 2023) and DINO(Caron et al. 2021), our baseline, here we use its training framework and apply our inference method to conduct comparative experiment. For a fair comparison, the saliency maps are either provided by the authors or generated by the offcially released models.

Quantitative Evaluation In Table 1, we present a comprehensive comparison of our proposed method with 7 state-of-the-art COD approaches on 4 widely-recognized datasets. The results clearly demonstrate the superior performance of SdalsNet. Specifically, our approach significantly surpasses all methods in CHAMELEON and COD10K. In CAMO and NC4K, excepted for S_α, which ranked second in the CAMO and third in the NC4K, and E_ϕ, which ranked third in the NC4K, the rest of the indicators were ranked the best.

Visual Evaluation We conduct a visual comparison with 7 state-of-the-art models, as depicted in Figure 3. For evaluation, we carefully selected six images from the four datasets where the objects are very hidden and the background is relatively prominent. By observing the discrimination between the foreground and background of the network, we find that the SdalsNet can find hidden objects from the complex background, while the rest of the network will be affected by the background, unable to find objects or even recognize the background as an object.

Method	CHAMELEON				CAMO				COD10K				NC4K			
	$S_\alpha \uparrow$	$E_\phi^{max} \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi^{max} \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi^{max} \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$E_\phi^{max} \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$
TokenCut	0.654	0.743	0.540	0.132	0.633	0.708	0.546	0.163	0.658	0.740	0.509	0.103	0.725	0.806	0.655	0.101
SelfMask	0.619	0.726	0.511	0.176	0.617	0.713	0.549	0.188	0.637	0.718	0.504	0.131	0.716	0.796	0.661	0.114
SpectralSeg	0.575	0.638	0.446	0.220	.579	0.486	0.658	0.235	0.575	0.606	0.395	0.193	0.669	0.729	0.570	0.159
A2S-v2	0.448	0.371	0.170	0.135	0.450	0.401	0.257	0.170	0.455	0.500	0.111	0.096	0.508	0.518	0.393	0.136
FOUND	0.684	0.812	0.591	0.095	0.685	0.784	0.635	0.129	0.670	0.753	0.521	0.085	0.741	**0.827**	0.676	**0.084**
UCOD-DA	0.715	0.804	0.631	0.095	**0.701**	0.786	0.647	0.127	0.689	0.741	0.548	0.086	**0.755**	0.822	0.691	0.085
DINO	0.649	0.765	0.553	0.123	0.636	0.747	0.570	0.152	0.624	0.694	0.463	0.109	0.684	0.771	0.610	0.113
SdalsNet	**0.724**	**0.837**	**0.666**	**0.08**	0.696	**0.801**	**0.664**	**0.117**	**0.696**	**0.781**	**0.572**	**0.071**	0.738	0.826	**0.697**	**0.084**

Table 1: Quantitative comparisons with SOTA UCOD methods on four widely using test datasets by four evaluation metrics. Top two performances are marked in Bold and Underline respectively. "↑"/"↓" indicates that larger or smaller is better.

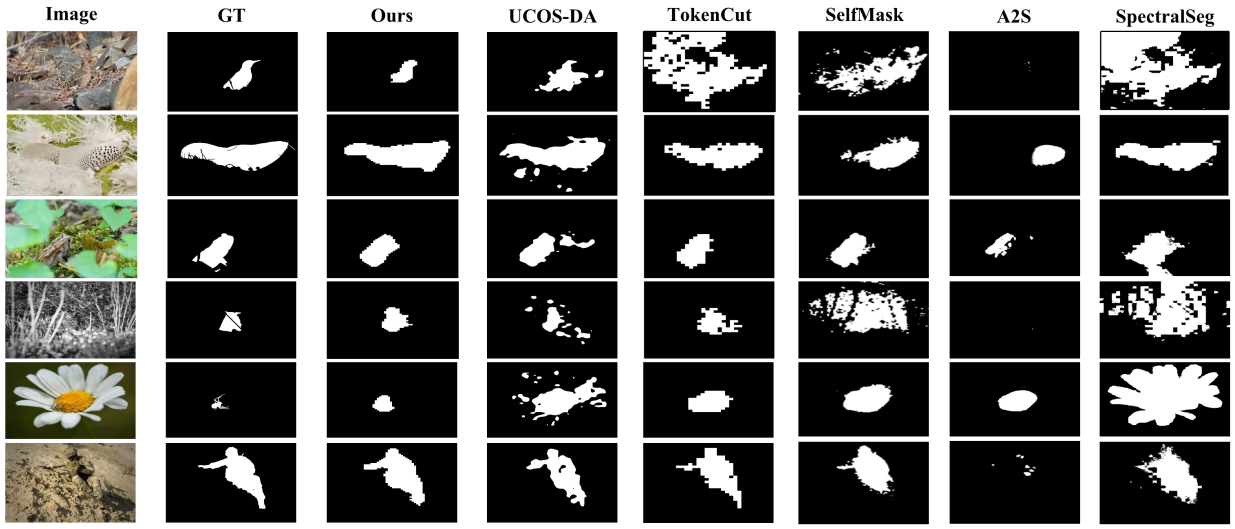


Figure 3: Qualitative comparison of our method with five state-of-the-arts on four datasets.

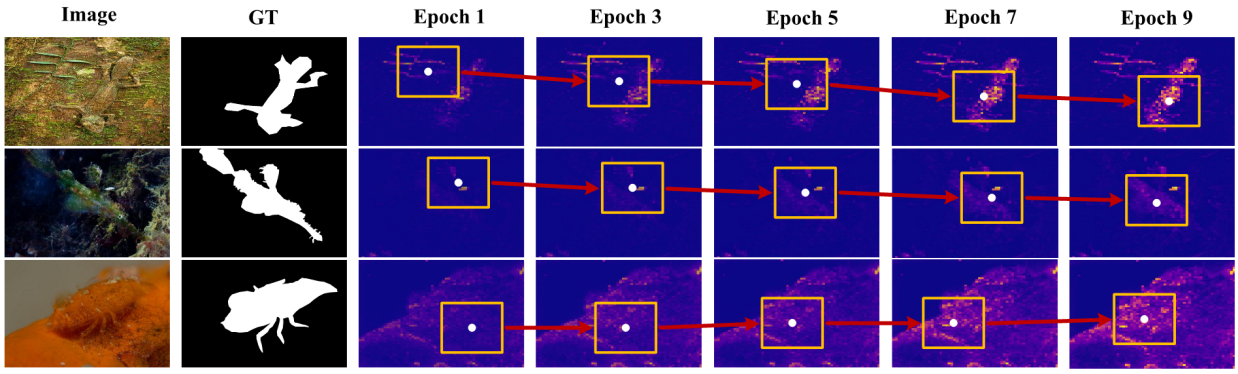


Figure 4: The attention maps of the attention shifting process at different epoch.

Ablation Study

Ablation Study on Modules The results of the ablation experiments for different modules are recorded in Table 2.

Firstly, we remove the Attention Localization Module and use the position corresponding to the maximum value in the attention map as the center of attention. According to the

ALM	ASL	CH	FT	CAMO			COD10K		
				$S_\alpha \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$
	✓	✓	✓	0.649	0.525	0.136	0.662	0.525	0.079
✓		✓	✓	0.681	0.634	0.127	0.673	0.535	0.083
✓	✓		✓	0.300	0.218	0.492	0.298	0.112	0.483
✓	✓	✓		0.646	0.581	0.147	0.633	0.469	0.106
✓	✓	✓	✓	**0.694**	**0.661**	**0.118**	**0.693**	**0.776**	**0.074**

Table 2: Ablation analysis of the modules on CAMO and COD10K datasets. ALM: Attention Localization Module. ASL: Attention Shift Learning. CH: Class-Head. FT: Frozen Training. The results of the full model are marked in Bold.

$\mathcal{L}_g$	$\mathcal{L}_f$	$\mathcal{L}_b$	$\mathcal{L}_p$	CAMO			COD10K		
				$S_\alpha \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{max} \uparrow$	$M \downarrow$
	✓	✓	✓	0.636	0.579	0.134	0.682	0.761	**0.069**
✓		✓	✓	0.616	0.539	0.166	0.598	0.415	0.129
✓	✓		✓	0.670	0.629	0.131	0.679	0.545	0.080
✓	✓	✓		0.677	0.636	0.129	0.679	0.548	0.079
✓	✓	✓	✓	**0.694**	**0.661**	**0.118**	**0.693**	**0.569**	0.074

Table 3: Ablation analysis of the loss function on CAMO and COD10K datasets. The results of the complete loss function are marked in Bold.

experimental results, simply using the maximum value to locate the attention center cannot get the center point well.

Next, we remove the Attention Shift Learning, keeping only the distillation loss $\mathcal{L}_g$ and remove the remaining three losses. Under such experimental settings, the attention of the network will not be shifted and promoted, and the segmentation results will become worse, and the above inferences are verified by experiments.

In fact, the above phenomena are intrinsically related. ALM provides accurate center points for ASL, which is a necessary prerequisite for ASL. ASL focuses attention on the center point, which will improve ALM localization. The two complement each other.

We then remove the Class-Head and directly use the output vectors of the ViT to compute the loss and train the network. The decrease in network performance is quite noticeable, as mapping ViT output vectors to higher dimensions using Class-Head can decode highly encoded features, contributing to the training process of the network.

Finally we remove the Frozen Training and let ViT and Class-Head train simultaneously. Experimental results show that Frozen Training can improve network performance.

Ablation Study on Loss Function The results of the ablation experiments for loss functions are recorded in Table 3. We take turns removing the four losses involved in this paper, including $\mathcal{L}_g$, $\mathcal{L}_f$, $\mathcal{L}_b$, and $\mathcal{L}_p$, without changing the network structure. It's evident that removing any single loss leads to a noticeable decline in accuracy compared to the complete loss. Evaluating the magnitude of decline establishes the hierarchy of importance among the four: $\mathcal{L}_f$ the ordering of the highest, achieving $29.87\%/14.27\%$ improvements in F_β^{max}/S_α, followed by $\mathcal{L}_g$, then the $\mathcal{L}_b$, the last is $\mathcal{L}_p$.

This conclusion shows that making the output corresponding to the foreground attention region close to the output of the teacher model can optimize the network attention and obtain better segmentation results. Relatively speaking, making the output, which corresponding to the background attention region, far away from the output, which corresponding to the foreground and global attention region can also optimize the attention of the network. The ablation experiments of $\mathcal{L}_b$ and $\mathcal{L}_p$ verify the above inferences.

Visualization Analysis of Attention Shift and Denoising

As shown in Figure 4, we visualize the center of attention and the surrounding area at each stage. We can see that as the training progresses, the center of attention is constantly moving toward the center of the object's region, and the attention is constantly focusing on the object. This shows that our network is constantly optimizing attention to focus on the object and further generating better attention centers. This process is repeated and eventually the network can iterate to a good result.

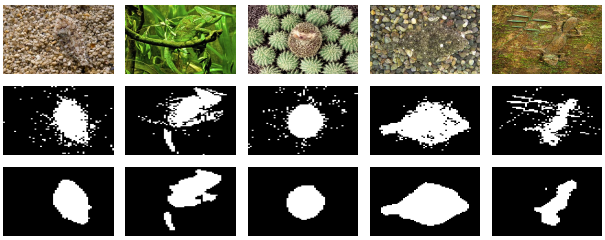


Figure 5: Images and binary maps before/after denoising.

As shown in Figure 5, we can find that under the action of the denoising method, the white small noise outside the target area is well removed, and the internal black noise is also well removed. Moreover, the denoising process does not destroy the holes inside the object or the concave regions at the edges. The objects are well segmented by the denoising method.

Conclusion

In this paper, we rethink the object detection task from the perspective of attention mechanism, and perform object detection indirectly by finding the area where the network's attention is concentrated. The first non-end-to-end UCOD model is designed, and the network is trained by attention localization and transfer to learn target features. In the future, we test the effect of this method in various object detection tasks and optimize the localization accuracy.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant 62206083.

References

Borji, A.; Cheng, M. M.; Jiang, H.; and Li, J. 2015. Salient Object Detection: A Benchmark. *IEEE Transactions on Image Processing*.

Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *International Conference on Computer Vision*.

Deng, J.; Dong, W.; Socher, R.; Li, L. J.; Li, K.; and Fei, L. F. 2009. A Large-Scale Hierarchical Image Database. In *Conference on Computer Vision and Pattern Recognition*.

Deng Ping, F.; Cheng, G.; Yang, C.; Bo, R.; Cheng, M. M.; and Borji, A. 2018. Enhanced-alignment Measure for Binary Foreground Map Evaluation. In *International Joint Conference on Artificial Intelligence*.

Deng Ping, F.; JI, G. P.; QIN, X.; and CHENG, M. 2021. Cognitive Vision Inspired Object Segmentation Metric and Loss Function. *Scientia Sinica Informationis*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *The Twelfth International Conference on Learning Representations*.

Fan, D. P.; Cheng, M. M.; Liu, Y.; Li, T.; and Borji, A. 2017. Structure-Measure: A New Way to Evaluate Foreground Maps. In *IEEE International Conference on Computer Vision*.

Fan, D. P.; Ji, G. P.; Cheng, M. M.; and Shao, L. 2022. Concealed Object Detection. *IEEE Trans on Pattern Analysis and Machine Intelligence*.

Fan, D. P.; Ji, G. P.; Sun, G.; Cheng, M. M.; Shen, J.; and Shao, L. 2020. Camouflaged object detection. In *Conference on Computer Vision and Pattern Recognition*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*.

Hou, Y.; Ma, Z.; Liu, C.; and Loy, C. C. 2019. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In *IEEE International Conference on Computer Vision*.

Hu, X.; Wang, S.; Qin, X.; Dai, H.; Ren, W.; Luo, D.; Tai, Y.; and Shao, L. 2023. High-Resolution Iterative Feedback Network for Camouflaged Object Detection. In *Association for the Advancement of Artificial Intelligence*.

Ji, G. P.; Fan, D. P.; Chou, Y. C.; Dai, D.; Liniger, A.; and Gool, L. V. 2023. Deep Gradient Learning for Efficient Camouflaged Object Detection. In *Machine Intelligence Research*.

Lea, T. N.; Nguyenb, T. V.; Nieb, Z.; Tranc, M.-T.; and Sugimotod, A. 2019. Computer Vision and Image Understanding. *Journal of Computer Vision and Image Understanding*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *International Conference on Computer Vision*.

Lv, Y.; Zhang, J.; Dai, Y.; Li, A.; Liu, B.; Barnes, N.; and Fan, D. P. 2021. Simultaneously Localize, Segment and Rank the Camouflaged Objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Melas-Kyriazi, L.; Rupprecht, C.; Laina, I.; and Vedaldi, A. 2022. Deep Spectral Methods: A Surprisingly Strong Baseline for Unsupervised Semantic Segmentation and Localization. In *Conference on Computer Vision and Pattern Recognition*.

Shin, G.; Albanie, S.; and Xie, W. 2023. Unsupervised Salient Object Detection with Spectral Cluster Voting. In *Conference on Computer Vision and Pattern Recognition Workshops*.

Siméoni, O.; Sekkat, C.; Puy, G.; Vobecky, A.; Éloi Zablocki; and Pérez, P. 2023. Unsupervised Object Localization: Observing the Background to Discover Objects. In *Conference on Computer Vision and Pattern Recognition*.

Skurowski, P.; Abdulameer, H.; Baszczyk, J.; Depta, T.; Kornacki, A.; and Kozie, P. 2017. Animal Camouflage Analysis: CHAMELEON Database. This work was led by Przemysław Skurowski. Gliwice, 2017.

Tan, M.; and Le, Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *International Conference on Machine Learning*.

Wang, Y.; Shen, X.; Hu, S. X.; Yuan, Y.; Crowley, J.; and Vaufreydaz, D. 2022. Self-Supervised Transformers for Unsupervised Object Discovery using Normalized Cut. In *Conference on Computer Vision and Pattern Recognition*.

Zagoruyko, S.; and Komodakis, N. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations*.

Zhang, Y.; and Wu, C. 2023. Unsupervised Camouflaged Object Segmentation as Domain Adaptation. In *International Conference on Computer Vision*.

Zhang, Y.; Zhang, J.; Hamidouche, W.; and Deforges, O. 2023. Predictive uncertainty estimation for camouflaged object detection. *IEEE Transactions on Image Processing*.

Zhou, H.; Qiao, B.; Yang, L.; Lai, J.; and Xie, X. 2023. Texture-Guided Saliency Distilling for Unsupervised Salient Object Detection. In *Conference on Computer Vision and Pattern Recognition*.

Zhu, H.; Li, P.; Xie, H.; Yan, X.; Liang, D.; Chen, D.; Wei, M.; and Qin, J. 2022. I Can Find You! Boundary-Guided Separated Attention Network for Camouflaged Object Detection. In *Association for the Advancement of Artificial Intelligence*.

Zhuge, M.; Fan, D. P.; Liu, N.; Zhang, D.; Xu, D.; and Shao, L. 2023. Salient Object Detection via Integrity Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.