ECAI 2023
K. Gal et al. (Eds.)

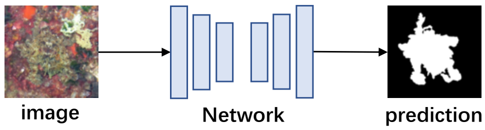
doi:10.3233/FAIA230302

Diffusion Model for Camouflaged Object Detection

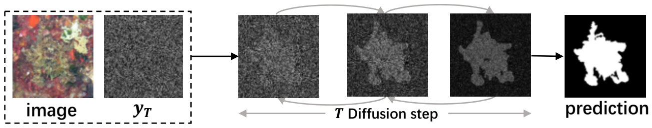
Zhennan Chen[a], Rongrong Gao[b], Tian-Zhu Xiang[c;*] and Fan Lin[a;*]

[a]School of Informatics, Xiamen University, Xiamen, China
[b]Department of Computer Science and Engineering, HKUST, Hong Kong, China
[c]G42, Abu Dhabi, UAE

Abstract. Camouflaged object detection is a challenging task that aims to identify objects that are highly similar to their background. Due to the powerful noise-to-image denoising capability of denoising diffusion models, in this paper, we propose a diffusion-based framework for camouflaged object detection, termed diffCOD, a new framework that considers the camouflaged object segmentation task as a denoising diffusion process from noisy masks to object masks. Specifically, the object mask diffuses from the ground-truth masks to a random distribution, and the designed model learns to reverse this noising process. To strengthen the denoising learning, the input image prior is encoded and integrated into the denoising diffusion model to guide the diffusion process. Furthermore, we design an injection attention module (IAM) to interact conditional semantic features extracted from the image with the diffusion noise embedding via the cross-attention mechanism to enhance denoising learning. Extensive experiments on four widely used COD benchmark datasets demonstrate that the proposed method achieves favorable performance compared to the existing 11 state-of-the-art methods, especially in the detailed texture segmentation of camouflaged objects. Our code will be made publicly available at: https://github.com/ZNan-Chen/diffCOD.



(a) Mainstream COD paradigm.



(b) Diffusion-based COD paradigm.

Figure 1: (a) The current mainstream COD paradigm inputs images into the network for prediction in a single direction, generating a deterministic segmentation mask. (b) Our proposed diffCOD provides a novel paradigm that decomposes COD into a series of forward-and-reverse diffusion processes.

1 Introduction

Camouflage is to use any combination of coloration, illumination, or materials to hide organisms in their surroundings, or disguise them as something else, for deception and paralysis purposes. Camouflaged object detection (COD) [10], that is, segmenting camouflaged objects from the background, is a challenging vision topic that has emerged in recent years, due to the high similarity of camouflaged objects to the background. COD has also attracted growing research interest from the computer vision community, because of its wide range of real-world applications, such as agricultural pest detection [22], medical image segmentation [26], and industrial defect detection [37].

With the advent of large-scale camouflaged object detection datasets in recent years, such as CAMO [23] and COD10K [10] datasets, numerous deep learning-based methods have been proposed and achieved great progress. Some methods are inspired by human visual mechanisms and adopt convolutional neural networks to imitate predation behavior, thus designing a series of models for COD, such as search identification network [9], positioning and focus network [28], zoom in and out [31], and PreyNet [45]. Some methods adopt auxiliary cues to improve network discrimination, or branch tasks to jointly learn camouflage features. The former typically employ frequency domain [47], edge/texture [18, 48], or motion information [4] to improve feature representation, and the latter usually introduces boundary detection [36], classification [23], fixation [27], or saliency detection [24] for multi-task collaborative learning. More recently, to improve global contextual exploration, transformer-based approaches have also been proposed, such as HitNet [16] and FSPNet [17]. Although these methods have greatly improved the performance of camouflaged object detection, the existing methods still struggle to achieve accurate location and segmentation in most complex scenarios, due to the interference of highly similar backgrounds and the complexity of the appearance of camouflaged objects.

In recent years, diffusion models [15] have demonstrated impressive performance in the generative modeling of images and videos [7], opening up a new era of generative models. Diffusion models are a class of generative models that consist of Markov chains trained using variational inference, to denoise noisy images blurred by Gaussian noise via learning the reverse diffusion process. Because of its powerful noise-to-image denoising pipeline, the computer vision community is curious about its variants for discriminative tasks [5]. More recently, diffusion models have been found to be highly effective in other computer vision tasks, such as image editing [14], super-resolution [25], instance segmentation [12], semantic segmentation [2, 3] and medical image segmentation [32, 39]. However, despite their great potential, diffusion models for challenging camouflaged object detection have still not been well explored.

In this paper, we propose to formulate the camouflaged object detection as a generative task, through a denoising diffusion process

* Corresponding Authors. Email: tianzhu.xiang19@gmail.com; iamafan@xmu.edu.cn.

from the noisy mask to the object mask in the image. Specifically, in the training stage, Gaussian noise is added to the ground-truth masks to obtain noisy masks, and then the model learns to reverse this noising process. In the inference stage, the model progressively refines a set of randomly generated noisy masks from the image through the learned denoising model, until they perfectly cover the targeted object without noise. We can see that the denoising diffusion model is the process of recovering the ground-truth mask from the random noisy distribution to the learned distribution over object masks. As shown in Figure 1, unlike previous deterministic network solutions that produce a single output for an input image, we decouple the detection of the object into a novel noise-to-mask paradigm with a series of forward-and-reverse diffusion steps, which can output masks from single or multi-step denoising, thereby generating multiple object segmentation masks from a single input image.

To this end, we propose a denoising diffusion-based model, termed diffCOD, which approaches camouflaged object tasks from the perspective of the noise-to-mask denoising diffusion process. The proposed model adopts a denoising network conditioned on the input image prior. The semantic features extracted from the image by a Transformer encoder are integrated into the denoising diffusion model to guide the diffusion process at each step. To effectively bridge the gap between the diffusion noise embedding and the conditional semantic features, an injection attention module (IAM) is designed to enhance the denoising diffusion learning by aggregating conditional semantic features and diffusion model encoder through a cross-attention mechanism. Our contributions are summarized as follows:

- We extend the denoising diffusion models to the task of camouflaged object detection, and propose a diffusion-based object segmentation model, called diffCOD, a novel framework that views camouflaged object detection as a denoising diffusion process from noisy masks to object masks.
- We design an injection attention module (IAM) to model the interaction between noise embeddings and image features. The proposed module adopts the cross-attention mechanism to integrate the conditional semantic feature extracted from the image into the diffusion model encoder to guide and enhance denoising learning.
- Extensive quantitative and qualitative experiments demonstrate that the proposed diffCOD achieves superior performance over the recent 11 state-of-the-art (SOTA) methods by a large margin, especially in object detail texture segmentation, indicating the effectiveness of the proposed method.

2 Related Work

2.1 Camouflaged Object Detection

Existing COD methods [8, 9, 10] are based on a non-generative approach to segment the objects from the background. The approaches in COD can be broadly categorized into the following strategies: a) Introducing additional cues to facilitate the exploration of camouflage features. BGNet [36] uses edge semantic information to enable the model to extract features that highlight the structure of the object and thus pinpoint the object boundary. TINet [48] designs a texture label to find boundaries and texture differences through progressive interactive guidance. FDCOD [47] incorporates frequency domain features into CNN models to better detect objects from the background. DGNet [18] utilizes gradient edge information to facilitate the generation of contextual and texture features. b) Multi-task learning strategies are used to improve segmentation capabilities. ANet [23] proposed joint learning of classification and segmentation tasks to help the model improve recognition accuracy. UJSC [24] detects both salient and camouflaged objects to improve the model performance. Rank-Net [27] proposes to use the localization model to find the obvious discriminative region of the camouflaged object, and the segmentation model to segment the full range of the camouflaged object. c) Coarse-to-fine feature learning strategy is utilized to explore and integrate multi-scale features. SegMaR [21] uses multi-stage detection to focus on the region where the goal is located. ZoomNet [30] learns multi-scale semantic information through multi-scale integration and hierarchical hybrid strategies to promote models that produce predictions with higher confidence. PreyNet [45] imitates the predation process for stepwise aggregation and calibration of features. PFNet [28] mimics nature's predation process by first locating potential targets from a global perspective and then gradually refining the fuzzy regions. SINet [10] is designed to improve segmentation performance by locating the object first and then differentiating the details. C^2FNet [35] proposes to use global contextual information to fuse on high-level features in a cascading manner to obtain better performance. HitNet [16] and FSPNet [17] propose to explore global context cues by transformers. In this paper, we introduce generative models, *i.e.*, denoising diffusion models, into the COD task to gradually refine the object masks from the noisy image, which achieve excellent performance, especially for objects with fine textures.

2.2 Diffusion Model

The diffusion model [15, 34] is a generative model that uses a forward Gaussian diffusion process to sample a noisy image, and then iteratively refines it using a backward generative process to obtain a denoised image. Diffusion models have shown strong potential in several fields, such as image synthesis [7, 15], image editing [14], and image super-resolution [6]. Moreover, the learning process of diffusion models is able to capture high-level semantic information that is valuable for segmentation tasks [2], which has led to a growing interest in diffusion models for image segmentation including medical image segmentation [39, 40], semantic segmentation [3, 20, 41, 43], and instance segmentation [1, 12]. MedSegDiff [39] proposes the first DPM-based medical segmentation model, and MedSegDiff-V2 [40] further improves the performance based on it using transformer. DDeP [3] finds that pre-training a semantic segmentation model as a denoising self-encoder is beneficial for performance improvement. DDP [20] designs a dense prediction framework with stepwise denoising refinement guided by image features. ODISE [43] combines a trained text image diffusion model with a discriminative model to achieve open-vocabulary panoptic segmentation. DiffuMask [41] uses a model for the automatic generation of image and pixel-level semantic annotations, and it also shows superiority in open vocabulary segmentation. DiffusionInst [12] proposes the first instance segmentation model based on a diffusion process to achieve global instance mask reconstruction. Segdiff [1] uses a diffusion probabilistic approach to design an end-to-end segmentation model that does not rely on a pre-trained backbone. However, there are no studies that demonstrate the effectiveness of diffusion models in COD tasks. In this work, we present the first diffusion model for the COD segmentation task.

3 Methodology

In this section, we first review the diffusion model (Sec. 3.1). Then we introduce the architecture of diffCOD (Sec. 3.2). Finally, we

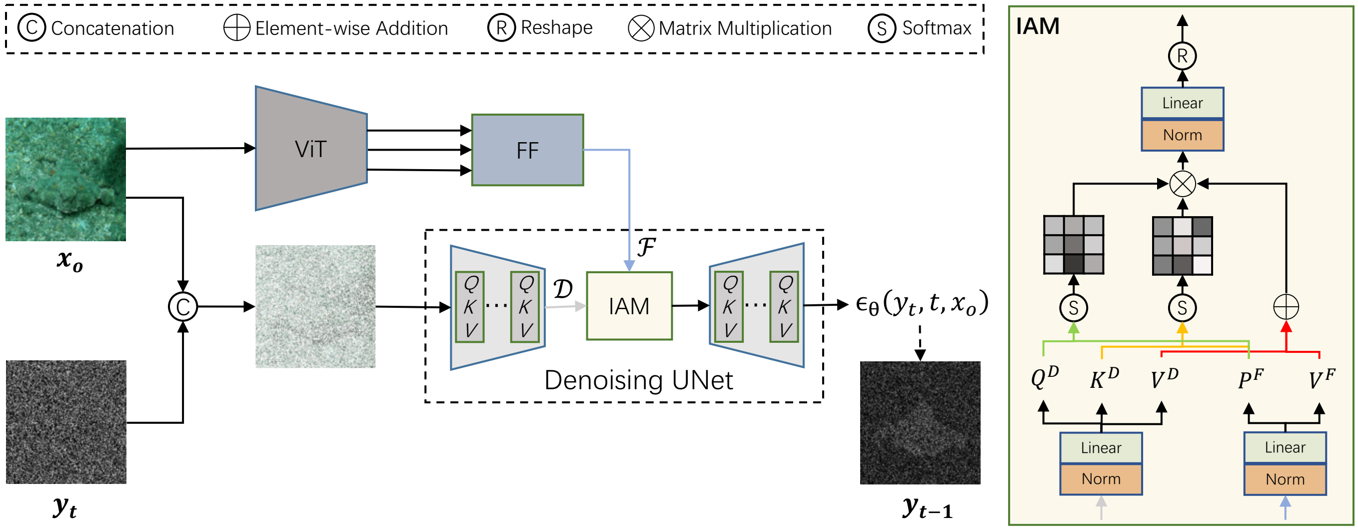


Figure 2: Our proposed diffCOD framework for COD, which feeds a given image into a denoising diffusion model with UNet architecture as the core component for denoising. An injection attention module (IAM) is designed to implicitly guide the diffusion process with the conditional semantic features that have gone through the backbone and feature fusion module (FF), allowing the model to take full advantage of the correspondence between image features and diffusion information.

describe the specific process of training and inference of diffCOD (Sec. 3.3 & Sec. 3.4).

3.1 Diffusion Model

The diffusion probability model has reaped plenty of attention due to its simple training process and excellent performance. It is mainly divided into forward process and reverse process. In the forward process, noise is added to the target image to make it closer to the Gaussian distribution. The reverse process learns to map the noise to the real image.

The forward process refers to the gradual incorporation of Gaussian noise with variance $\beta_t \in (0,1)$ into the original image $x_0 \sim p(x_0)$ at time t until it converges to isotropic Gaussian distribution. The forward process is described by the formulation:

$$q\left(x_t \mid x_{t-1}\right)=\mathcal{N}\left(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathbf{I}\right) \quad (1)$$

where $t \in [1, T]$. We can obtain the latent variable x_t directly by using x_0 by the following equation:

$$q\left(x_t \mid x_0\right)=\mathcal{N}\left(x_t ; \sqrt{\bar{\alpha}_t} x_0,\left(1-\bar{\alpha}_t\right) \mathbf{I}\right) \quad (2)$$

where $\alpha_t := 1-\beta_t$, $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

The reverse process converts the latent variable distribution $p(x_T)$ to $p(x_0)$ through a Markov chain, and the reverse process can be denoted as follows:

$$p_\theta\left(x_{t-1} \mid x_t\right)=\mathcal{N}\left(x_{t-1} ; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \quad (3)$$

The combination of q and p is a variational auto-encoder, and the variational lower bound (VLB) is defined as follows:

$$L_{\mathrm{vlb}} := L_0+L_1+\ldots+L_{T-1}+L_T \quad (4)$$

$$L_0 := -\log p_\theta\left(x_0 \mid x_1\right) \quad (5)$$

$$L_{t-1} := D_{KL}\left(q\left(x_{t-1} \mid x_t, x_0\right) \| p_\theta\left(x_{t-1} \mid x_t\right)\right) \quad (6)$$

$$L_T := D_{KL}\left(q\left(x_T \mid x_0\right) \| p\left(x_T\right)\right) \quad (7)$$

3.2 Architecture

As shown in Figure 2, the proposed diffCOD aims to solve the COD task by the diffusion model. The denoising network of diffCOD is based on the UNet architecture [33]. To get effective conditional semantic features, we obtain multi-scale features by ViT-based backbone and feature fusion (FF) to yield features containing rich multi-scale details. In addition, to let the texture patterns and localization information in the conditional semantic features guide the denoising process, we propose an injection attention module (IAM) based on cross-attention. This allows the network to reduce the difference between diffusion features and image features and to combine the advantages of both.

Feature Fusion (FF). Given an initial input image $x_o \in \mathbb{R}^{H\times W\times 3}$, we adopt the top-three high-level features of the visual backbone as our multi-scale backbone features, denoted as $\mathcal{X}_i^p$, $i \in \{1,2,3\}$ whose resolution is $\frac{H}{k} \times \frac{W}{k}$, $k \in \{8,16,32\}$. Here we use PVTv2 [38] as the backbone. Then FF is used to aggregate these multiscale features. Specifically, FF contains three branches to process $\mathcal{X}_i^p$, each branch uses two convolution operations with 3×3 kernel for feature enhancement, and finally the three branches are coalesced by a single convolution to obtain $\mathcal{F} \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C}$.

Injection Attention Module (IAM). To introduce texture and location information of the original features in the noise prediction process, we employ a cross-attention-based IAM, which is embedded in the middle of the UNet-based denoising network. Given the multiscale fusion feature $\mathcal{F}$ from FF and the deepest feature $\mathcal{D} \in \mathbb{R}^{\frac{H}{32}\times\frac{W}{32}\times C}$ from the diffusion model as the common input to the IAM. Specifically, $\mathcal{D}$ is transformed by linear projection to generate the query $\mathbf{Q^D}$, the key $\mathbf{K^D}$ and the value $\mathbf{V^D}$. $\mathcal{F}$ generates $\mathbf{P^F}$, $\mathbf{V^F}$ by linear projection, and it is noteworthy that $\mathcal{F}$ does not generate the query and the key for similarity comparison, but uses the generated $\mathbf{P^F}$ to act as an intermediary for similarity comparison with $\mathcal{D}$. This process is defined as follows:

$$\mathbf{Q}^{\mathbf{D}} = \mathcal{D} \cdot \mathcal{W}_{\mathcal{Q}}^{\mathcal{D}}, \quad \mathbf{K}^{\mathbf{D}} = \mathcal{D} \cdot \mathcal{W}_{\mathcal{K}}^{\mathcal{D}}, \quad \mathbf{V}^{\mathbf{D}} = \mathcal{D} \cdot \mathcal{W}_{\mathcal{V}}^{\mathcal{D}} \\ \mathbf{P}^{\mathbf{F}} = \mathcal{F} \cdot \mathcal{W}_{\mathcal{P}}^{\mathcal{F}}, \quad \mathbf{V}^{\mathbf{F}} = \mathcal{F} \cdot \mathcal{W}_{\mathcal{V}}^{\mathcal{F}} \tag{8}$$

where $\mathcal{W}_{\mathcal{Q}}^{\mathcal{D}}, \mathcal{W}_{\mathcal{K}}^{\mathcal{D}}, \mathcal{W}_{\mathcal{V}}^{\mathcal{D}}, \mathcal{W}_{\mathcal{P}}^{\mathcal{F}}, \mathcal{W}_{\mathcal{V}}^{\mathcal{F}} \in \mathbb{R}^{d \times d}$. d is the dimensionality.

Thus the IAM operation is defined as follows:

$$\mathbf{M}_1^{att} = \text{Softmax}\left(\frac{\mathbf{Q}^{\mathbf{D}} \cdot (\mathbf{P}^{\mathbf{F}})^T}{\sqrt{d}}\right) \tag{9}$$

$$\mathbf{M}_2^{att} = \text{Softmax}\left(\frac{\mathbf{K}^{\mathbf{D}} \cdot (\mathbf{P}^{\mathbf{F}})^T}{\sqrt{d}}\right) \tag{10}$$

$$O^I = \mathbf{M}_1^{att} \cdot \mathbf{M}_2^{att} \cdot (\mathbf{V}^D + \mathbf{V}^F) \tag{11}$$

where $\mathbf{M}_1^{att}$ and $\mathbf{M}_2^{att}$ represent the attention maps of $\mathbf{Q}^{\mathbf{D}}$-$\mathbf{P}^{\mathbf{F}}$ and $\mathbf{K}^{\mathbf{D}}$-$\mathbf{P}^{\mathbf{F}}$, respectively. $O^I \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C}$ denotes the final generated cross-attention fusion feature.

3.3 Training

In the forward process, the Gaussian noise ϵ_t is added to the ground truth y_0 to obtain the noise mapping $y_t \sim q(y_t \mid y_0)$ by T-steps. The intensity of the noise is controlled by α_t and conforms to the standard normal distribution. This process can be defined as follows:

$$y_t = \sqrt{\alpha_t} y_{t-1} + (1 - \alpha_t)\, \epsilon_t \tag{12}$$

where $t = [1, \cdots, T]$ and $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$.

By iterative computation, we can directly obtain y_t. This process can be further marginalized as:

$$y_t = \sqrt{\bar{\alpha}_t} y_0 + (1 - \bar{\alpha}_t)\, \epsilon_t \tag{13}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$.

In the reverse process, we map from y_t to y_{t-1} until the segmented image is acquired step by step. The mathematics is defined as follows:

$$y_{t-1} = \mu_\theta(y_t, t, x_o) + \Sigma_\theta(y_t, t, x_o)\, \epsilon_t \tag{14}$$

We train a denoising UNet model to predict $\epsilon_\theta(y_t, t, x_o)$:

$$\mu_\theta(y_t, t, x_o) = \frac{\left(y_t - \left(\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\right) \epsilon_\theta(y_t, t, x_o)\right)}{\sqrt{\alpha_t}} \tag{15}$$

We follow the improved DDPM [29] to simplify Eq. (4)-(7) to define the hybrid objective $L_{\text{hybrid}} = L_{\text{simple}} + L_{\text{vlb}}$. L_{vlb} learns the term $\Sigma_\theta(y_t, t, x_o)$. Furthermore, inspired by [40], we use FF and a convolution layer to provide an initial static mask y_m to reduce the diffusion variance, and its mean square loss is defined as L_{static}. Total loss function L_{total} is defined as follows:

$$\begin{cases} L_{\text{simple}} & = \mathbb{E}_{t \sim [1,T], y_0 \sim q(y_0), \epsilon} \left\| \epsilon - \epsilon_\theta(y_t, t, x_o) \right\|^2 \\ L_{\text{static}} & = \mathbb{E}_{y_0 \sim q(y_0), y_m} \left\| y_0 - y_m \right\|^2 \\ L_{\text{total}} & = L_{\text{simple}} + L_{\text{vlb}} + L_{\text{static}} \end{cases} \tag{16}$$

Algorithm 1 provides the training procedure for diffCOD.

3.4 Inference

In the inference stage, we step-by-step apply Eq. (14) to sample a pure Gaussian noise $y_t \sim \mathcal{N}(0, I)$. In addition, we add conditional information related to the image features to guide the inference process. After performing T iterations, we can obtain the segmentation image of the camouflaged object. Using the setting of [29] for the sampling, the inference process of diffCOD is shown in Algorithm 2.

Algorithm 1: diffCOD Training

```
def training_loss(images, masks):
  """images: [b, h, w, 3], masks: [b, h, w, 1]"""

  # Encode images
  X_p = ViT(images)
  F = FF(X_p)

  # corrupt groundtruth
  t = uniform(0, 1)
  eps = normal(mean=0, std=1)
  mask_crpt = sqrt(gamma(t)) * masks +
              sqrt(1 - gamma(t)) * eps

  # predict and backward
  D = UNet_1(images, mask_crpt, t)
  O = IAM(F, D)
  preds = UNet_2(O)

  # compute loss
  loss = loss_function(preds, masks)
  return loss
```

Algorithm 2: diffCOD Inference

```
def inference(images, steps):
  """images: [b, h, w, 3], steps: sample steps"""

  # Encode images
  X_p = ViT(images)
  F = FF(X_p)

  m_t = normal(mean=0, std=1)

  # time intervals
  for step in range(steps):
    out = p_sample(images, F, m_t, step)

  return out
```

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on four widely used benchmark datasets of COD task, *i.e.*, CAMO , CHAMELEON, COD10K and NC4K. The details of each dataset are as follows:

- CAMO contains 1,250 camouflaged images and 1,250 non-camouflaged images, covering eight categories.
- CHAMELEON has a total of 76 camouflaged images.
- COD10K consists of 5,066 camouflaged, 1,934 non-camouflaged, and 3,000 background images. It is currently the largest dataset which covers 10 superclasses and 78 subclasses.
- NC4K is a newly published dataset that has a total of 4,121 camouflaged images.

Following the standard practice of COD tasks, we use 3,040 images from COD10K and 1,000 images from CAMO as the training set and the remaining data as the test set.

Evaluation metrics. According to the standard evaluation protocol of COD, we employ the five common metrics to evaluate our model, *i.e.*, structure-measure (S_α), weighted F-measure (F_β^ω), mean F-measure (F_m), mean E-measure (E_m) and mean absolute error (MAE). The purpose of structure-measure (S_α) is to evaluate the

Method	COD10K					NC4K					CAMO					CHAMELEON				
	$S_\alpha \uparrow$	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$	S_α	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$	S_α	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$
2019 CPD [42]	0.736	0.547	0.607	0.801	0.053	0.769	0.652	0.713	0.822	0.072	0.688	0.552	0.623	0.728	0.114	0.876	0.809	0.821	0.914	0.036
2019 EGNet [46]	0.746	0.560	0.591	0.789	0.053	0.804	0.727	0.731	0.834	0.066	0.730	0.579	0.693	0.762	0.104	0.851	0.705	0.747	0.869	0.049
2020 SINet [10]	0.772	0.543	0.640	0.810	0.051	0.810	0.665	0.741	0.841	0.066	0.753	0.602	0.676	0.774	0.097	0.867	0.727	0.792	0.889	0.044
2020 MINet [31]	0.780	0.628	0.677	0.838	0.040	0.810	0.717	0.764	0.856	0.057	0.741	0.629	0.682	0.783	0.096	0.853	0.768	0.803	0.902	0.035
2020 PraNet [11]	0.800	0.656	0.699	0.869	0.041	0.826	0.739	0.780	0.878	0.056	0.769	0.664	0.716	0.812	0.091	0.870	0.790	0.816	0.915	0.039
2021 PFNet [28]	0.797	0.656	0.698	0.875	0.039	0.826	0.743	0.783	0.884	0.054	0.774	0.683	0.737	0.832	0.087	0.889	0.823	**0.840**	**0.946**	**0.030**
2021 LSR [27]	0.805	0.660	0.703	0.876	0.039	0.832	0.743	0.785	0.888	0.053	0.793	0.703	0.753	0.850	0.083	0.890	0.824	0.834	0.932	0.034
2022 ERRNet [19]	0.780	0.629	0.679	0.867	0.044	—	—	—	—	—	0.761	0.660	0.719	0.817	0.088	0.877	0.805	0.821	0.927	0.036
2022 NCHIT [44]	0.790	0.608	0.689	0.817	0.046	—	—	—	—	—	0.780	0.671	0.733	0.803	0.088	0.874	0.793	0.812	0.891	0.041
2022 CubeNet [49]	0.795	0.644	0.681	0.864	0.041	—	—	—	—	—	0.788	0.682	0.743	0.838	0.085	0.873	0.787	0.823	0.928	0.037
2023 CRNet [13]	0.733	0.576	0.627	0.832	0.049	—	—	—	—	—	0.735	0.641	0.702	0.815	0.092	0.818	0.744	0.756	0.897	0.046
diffCOD	**0.812**	**0.684**	**0.723**	**0.892**	**0.036**	**0.837**	**0.761**	**0.802**	**0.891**	**0.051**	**0.795**	**0.704**	**0.758**	**0.852**	**0.082**	**0.893**	**0.826**	0.837	0.933	**0.030**

Table 1: Quantitative comparisons of our proposed method and other 11 state-of-the-art methods on four widely used benchmark datasets. The higher the S_α, F_β^ω, F_m, and E_m, the better the performance. The smaller the MAE, the better. The best results are marked in **bold**.

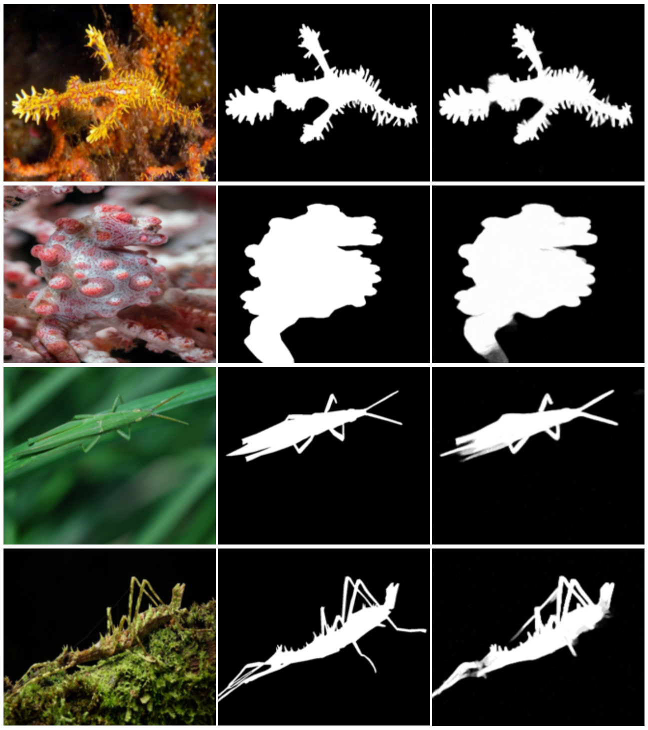
structural information of the result and ground truth, including object perception and region perception. Weighted F-measure F_β^ω is the weighted information of the mean F-measure (F_m) metric, and these two metrics are a combined assessment of the accuracy and recall of the result. Mean E-measure (E_m) is able to perform both pixel-level matching and image-level statistics, and is used to calculate the overall and local accuracy of the segmentation results. The mean absolute error (MAE) metric is often used to evaluate the average pixel-level relative error between the result and ground truth.

Implementation details. The proposed method is implemented with the PyTorch toolbox. We set the time step as T = 1000 with a linear noise schedule for all the experiments. We use Adam as our model optimizer with a learning rate of 1e-4. The batch size is set to 64. During the training, the input images are resized to 256×256 via bilinear interpolation and augmented by random flipping, cropping, and color jittering.

Baselines. Our diffCOD is compared with 11 recent state-of-the-art methods, including CPD [42], EGNet [46], SINet [10], MINet [31], PraNet [11], PFNet [28], LSR [27], ERRNet [19], NCHIT [44], CubeNet [49], CRNet [13]. For a fair comparison, all results are either provided by the authors or reproduced by an open-source model re-trained on the same training set with the recommended setting.

4.2 *Quantitative Evaluation*

The quantitative comparison of our proposed diffCOD with 11 state-of-the-art methods is shown in Table 1. Our method achieves superior performance over other competitors, indicating that our model can generate high-quality camouflaged segmentation masks compared to previous methods. For the largest COD10K dataset, our method shows a substantial performance jump, with an average increase of 4.8%, 12.8%, 9.5%, 6.4% and 19.1% for S_α, F_β^ω, F_m, E_m and MAE, respectively. For another recent large-scale NC4K dataset, diffCOD also outperforms all methods, increasing by 3.4%, 7.1%, 6.1%, 4.0% and 14.8% on average for S_α, F_β^ω, F_m, E_m and MAE, respectively. In addition, the most significant increases in the CAMO dataset were seen in the F_β^ω and MAE, with improvements of 10.2% and 11.3%, respectively. CHAMELEON is the smallest COD dataset, therefore most of the methods perform inconsistently on this dataset, our method increases 3.0%, 6.2%, 4.0%, 2.6% and 21.2% for S_α, F_β^ω, F_m, E_m and MAE, respectively.



(a) Image (b) GT (c) diffCOD

Figure 3: Visual results of our proposed model in terms of detailed textures.

4.3 *Qualitative Evaluation*

Figure 4 shows a comprehensive visual comparison with current state-of-the-art methods. It can be found that our method achieves competitive visual performance in different types of challenging scenarios. Our diffCOD is able to guarantee the integrity and correctness of recognition even under difficult conditions, such as single object (*e.g.,* row 1-4), multi-objects (*e.g.,* row 5-8), small object (*e.g.,* row 9-11). Nature's camouflaged organisms often have strange traits, such as tentacles, tiny spikes, etc. Past models have blurred the recognition of edge parts even if the location of the target is correctly targeted. However, we are surprised by the advantages of diffCOD in terms of detailed textures. As shown in Figure 3, our method is able to

Figure 4: Qualitative comparison of our proposed method and other representative COD methods. Our method provides better performance than all competitors for camouflaged object segmentation in various complex scenes.

accurately identify every subtlety, and it can depict the textures of the object in extremely fine detail, solving the blurring problem of segmentation masks in other methods.

4.4 Ablation Studies

Overview. We perform ablation studies on key components to verify their effectiveness and analyze their impacts on performance, as shown in Table 2. Experimental results demonstrate that our designed Injection Attention Module (IAM), Feature Fusion (FF), and ViT can improve detection performance. When they are combined to build diffCOD, significant improvements in all evaluation metrics are observed. Note that the Baseline refers to the standard diffusion model.

Effectiveness of IAM. As can be seen in Table 2, the presence or absence of IAM plays a key role in the performance improvement of the model. Compared to the experiments without this key component, the average improvement of #2 with IAM over #1 for S_α, F_β^ω, F_m,

No.	Component				COD10K					NC4K					CAMO				
	Baseline	IAM	FF	ViT	$S_\alpha \uparrow$	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$	$S_\alpha \uparrow$	$F_\beta^\omega \uparrow$	$F_m \uparrow$	$E_m \uparrow$	$MAE \downarrow$
#1	✓				0.761	0.604	0.657	0.845	0.046	0.781	0.687	0.712	0.841	0.061	0.731	0.607	0.664	0.790	0.097
#2	✓	✓			0.788	0.638	0.687	0.861	0.041	0.805	0.711	0.747	0.863	0.056	0.749	0.631	0.694	0.805	0.093
#3	✓	✓	✓		0.801	0.662	0.709	0.876	0.039	0.823	0.731	0.772	0.876	0.054	0.770	0.664	0.718	0.829	0.087
#4	✓	✓		✓	0.809	0.677	0.719	0.888	0.036	0.835	0.758	0.798	0.889	0.051	0.792	0.693	0.751	0.849	0.083
#5	✓		✓	✓	0.799	0.657	0.708	0.868	0.039	0.820	0.727	0.770	0.872	0.054	0.772	0.663	0.722	0.831	0.086
#OUR	✓	✓	✓	✓	**0.812**	**0.684**	**0.723**	**0.892**	**0.036**	**0.837**	**0.761**	**0.802**	**0.891**	**0.051**	**0.795**	**0.704**	**0.758**	**0.852**	**0.082**

Table 2: Ablation studies of our diffCOD. The best results are marked in **bold**.

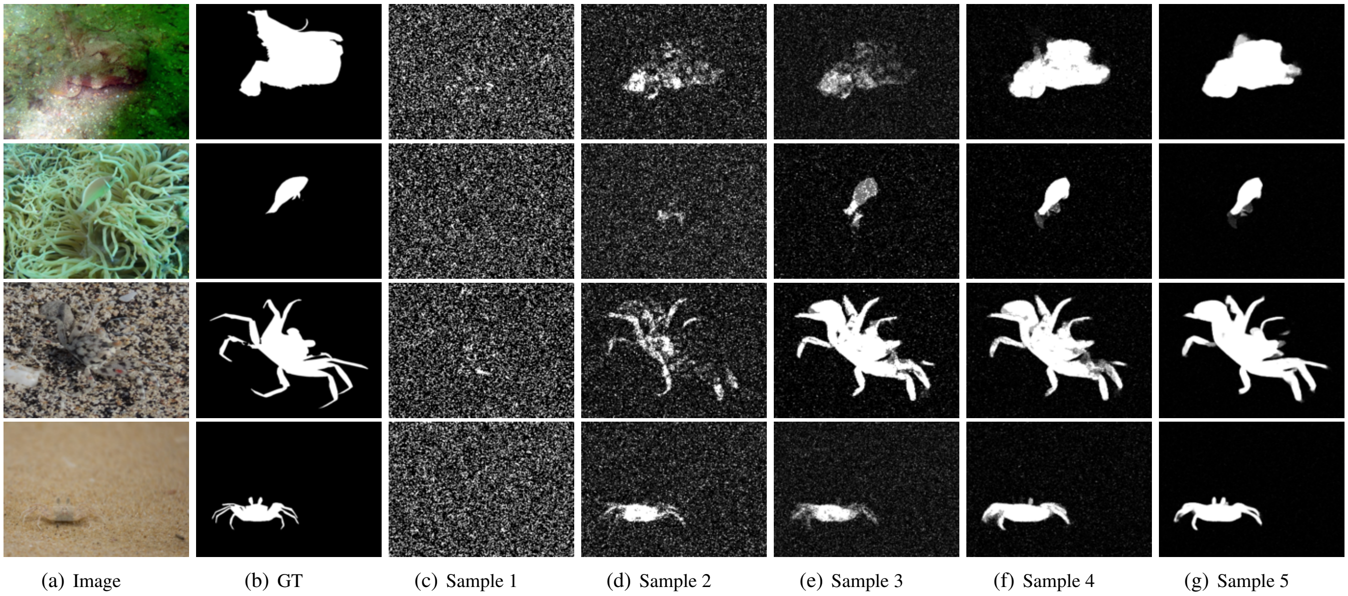


Figure 5: Visual results of the sampling process. (c)-(g) is the diffCOD sampling process. The time step is 200, 400, 600, 800, and 1000, respectively.

E_m and MAE on the three datasets is 3.0%, 4.3%, 4.7%, 2.1% and 7.7%, respectively. Furthermore, #Our accuracy improvement over #5 is significant, with an average increase of 6.0% in MAE metric on the three datasets. This is a good indication that IAM integrates diffusion features and texture features from the backbone perfectly.

Effectiveness of FF. The main role of FF is to aggregate the multi-scale features. As shown in Table 2, compared to No. #2, No. #3 has an average improvement of 2.2%, 5.0%, 3.8%, 2.5% and 6.0% for S_α, F_β^ω, F_m, E_m and MAE on the three datasets, respectively. The performance of #Ours on S_α, F_β^ω, F_m and E_m is 3.2%, 1.0%, 0.7% and 0.3% higher than that of No. #4.

Effectiveness of ViT. To obtain the location information and texture information of the objects in the original features, we use a ViT as a backbone to assist the diffusion process. From Table 2, we can learn that #Ours containing rich original features has an average improvement of 2.1%, 4.5%, 3.8%, 2.1% and 6.3% over #3 for S_α, F_β^ω, F_m, E_m and MAE on the three datasets, respectively. #2, which contains no original features at all, has an average of 4.0%, 7.5%, 6.6%, 3.9% and 10.6% lower than #4 for S_α, F_β^ω, F_m, E_m and MAE on the three data sets, respectively. In addition, to further demonstrate the significance of conditional semantic features to guide the diffusion process, we visualize the sampling process of diffCOD. From Figure 5, we can see that our model learns part of the location information and texture patterns of the camouflaged objects at the early stage of denoising, and the subsequent inference process gradually refines the final mask by training out the denoising model on this basis. This shows that the key clues extracted by ViT are perfectly integrated into the diffusion process with the help of FF and IAM.

5 Conclusion

In this paper, we propose a diffusion-based framework for camouflaged object detection, which changes the previous detection paradigm of the COD community by using a generative model for the segmentation of camouflaged objects to achieve significant performance gains. To the best of our knowledge, this is the first framework that employs a denoising diffusion model for COD tasks. Our approach decouples the task of segmenting camouflaged objects into a series of forward and reverse diffusion processes, and integrates key information from conditional semantic features to guide this process. Extensive experiments show the superiority over 11 other state-of-the-art methods on four datasets. As a new paradigm for camouflaged object detection, we hope that our proposed method will serve as a solid baseline and encourage future research.

References

[1] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf, 'Segdiff: Image segmentation with diffusion probabilistic models', *arXiv preprint arXiv:2112.00390*, (2021).

[2] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko, 'Label-efficient semantic segmentation with diffusion models', in *ICLR*, (2022).

[3] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi, 'Denoising pretraining for semantic segmentation', in *CVPR*, pp. 4175–4186, (2022).

[4] Xuelian Cheng, Huan Xiong, Deng-Ping Fan, Yiran Zhong, Mehrtash Harandi, Tom Drummond, and Zongyuan Ge, 'Implicit motion handling for video camouflaged object detection', in *CVPR*, pp. 13864–13873, (2022).

[5] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah, 'Diffusion models in vision: A survey', *IEEE TPAMI*, (2023).

[6] Max Daniels, Tyler Maunu, and Paul Hand, 'Score-based generative neural networks for large-scale optimal transport', *NeurIPS*, **34**, 12955–12965, (2021).

[7] Prafulla Dhariwal and Alexander Nichol, 'Diffusion models beat gans on image synthesis', *NeurIPS*, **34**, 8780–8794, (2021).

[8] Bo Dong, Jialun Pei, Rongrong Gao, Tian-Zhu Xiang, Shuo Wang, and Huan Xiong, 'A unified query-based paradigm for camouflaged instance segmentation', in *ACM MM*, (2023).

[9] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao, 'Concealed object detection', *IEEE TPAMI*, (2021).

[10] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao, 'Camouflaged object detection', in *CVPR*, pp. 2777–2787, (2020).

[11] 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 *MICCAI*, pp. 263–273. Springer, (2020).

[12] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang, 'Diffusioninst: Diffusion model for instance segmentation', *arXiv preprint arXiv:2212.02773*, (2022).

[13] Ruozhen He, Qihua Dong, Jiaying Lin, and Rynson WH Lau, 'Weakly-supervised camouflaged object detection with scribble annotations', *AAAI*, 781–789, (2023).

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or, 'Prompt-to-prompt image editing with cross attention control', *arXiv preprint arXiv:2208.01626*, (2022).

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel, 'Denoising diffusion probabilistic models', *NeurIPS*, **33**, 6840–6851, (2020).

[16] 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', *AAAI*, (2023).

[17] 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', *CVPR*, (2023).

[18] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool, 'Deep gradient learning for efficient camouflaged object detection', *Machine Intelligence Research*, (2023).

[19] Ge-Peng Ji, Lei Zhu, Mingchen Zhuge, and Keren Fu, 'Fast camouflaged object detection via edge-based reversible re-calibration network', *Pattern Recognition*, **123**, 108414, (2022).

[20] Yuanfeng Ji, Zhe Chen, Enze Xie, Lanqing Hong, Xihui Liu, Zhaoqiang Liu, Tong Lu, Zhenguo Li, and Ping Luo, 'Ddp: Diffusion model for dense visual prediction', *arXiv preprint arXiv:2303.17559*, (2023).

[21] 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*, pp. 4713–4722, (2022).

[22] Karthika Suresh Kumar and Aamer Abdul Rahman, 'Early detection of locust swarms using deep learning', in *Advances in machine learning and computational intelligence*, 303–310, Springer, (2021).

[23] Trung-Nghia Le, Tam V Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto, 'Anabranch network for camouflaged object segmentation', *CVIU*, **184**, 45–56, (2019).

[24] Aixuan Li, Jing Zhang, Yunqiu Lv, Bowen Liu, Tong Zhang, and Yuchao Dai, 'Uncertainty-aware joint salient object and camouflaged object detection', in *CVPR*, pp. 10071–10081, (2021).

[25] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen, 'Srdiff: Single image super-resolution with diffusion probabilistic models', *Neurocomputing*, **479**, 47–59, (2022).

[26] Lin Li, Jingyi Liu, Shuo Wang, Xunkun Wang, and Tian-Zhu Xiang, 'Trichomonas vaginalis segmentation in microscope images', in *MICCAI*, pp. 68–78. Springer, (2022).

[27] 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*, pp. 11591–11601, (2021).

[28] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan, 'Camouflaged object segmentation with distraction mining', in *CVPR*, pp. 8772–8781, (2021).

[29] Alexander Quinn Nichol and Prafulla Dhariwal, 'Improved denoising diffusion probabilistic models', in *ICML*, pp. 8162–8171, (2021).

[30] 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*, pp. 2160–2170, (2022).

[31] Youwei Pang, Xiaoqi Zhao, Lihe Zhang, and Huchuan Lu, 'Multi-scale interactive network for salient object detection', in *CVPR*, pp. 9413–9422, (2020).

[32] Aimon Rahman, Jeya Maria Jose Valanarasu, Ilker Hacihaliloglu, and Vishal M Patel, 'Ambiguous medical image segmentation using diffusion models', in *CVPR*, (2023).

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, 'U-net: Convolutional networks for biomedical image segmentation', in *MICCAI*, pp. 234–241. Springer, (2015).

[34] Yang Song and Stefano Ermon, 'Generative modeling by estimating gradients of the data distribution', *NeurIPS*, **32**, (2019).

[35] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu, 'Context-aware cross-level fusion network for camouflaged object detection', *IJCAI*, 1025–1031, (2021).

[36] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang, 'Boundary-guided camouflaged object detection', *IJCAI*, 1335–1341, (2022).

[37] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj, 'Segmentation-based deep-learning approach for surface-defect detection', *Journal of Intelligent Manufacturing*, **31**(3), 759–776, (2020).

[38] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao, 'Pvtv2: Improved baselines with pyramid vision transformer', *Computational Visual Media*, **8**(3), 1–10, (2022).

[39] Junde Wu, Huihui Fang, Yu Zhang, Yehui Yang, and Yanwu Xu, 'Medsegdiff: Medical image segmentation with diffusion probabilistic model', *MIDL*, (2023).

[40] Junde Wu, Rao Fu, Huihui Fang, Yu Zhang, and Yanwu Xu, 'Medsegdiff-v2: Diffusion based medical image segmentation with transformer', *arXiv preprint arXiv:2301.11798*, (2023).

[41] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen, 'Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models', *ICCV*, (2023).

[42] Zhe Wu, Li Su, and Qingming Huang, 'Cascaded partial decoder for fast and accurate salient object detection', in *CVPR*, pp. 3907–3916, (2019).

[43] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello, 'Open-vocabulary panoptic segmentation with text-to-image diffusion models', *CVPR*, (2023).

[44] Cong Zhang, Kang Wang, Hongbo Bi, Ziqi Liu, and Lina Yang, 'Camouflaged object detection via neighbor connection and hierarchical information transfer', *CVIU*, **221**, 103450, (2022).

[45] Miao Zhang, Shuang Xu, Yongri Piao, Dongxiang Shi, Shusen Lin, and Huchuan Lu, 'Preynet: Preying on camouflaged objects', in *ACM MM*, pp. 5323–5332, (2022).

[46] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng, 'Egnet: Edge guidance network for salient object detection', in *ICCV*, pp. 8778–8787, (October 2019).

[47] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding, 'Detecting camouflaged object in frequency domain', in *CVPR*, pp. 4504–4513, (2022).

[48] Jinchao Zhu, Xiaoyu Zhang, Shuo Zhang, and Junnan Liu, 'Inferring camouflaged objects by texture-aware interactive guidance network', in *AAAI*, pp. 3599–3607, (2021).

[49] Mingchen Zhuge, Xiankai Lu, Yiyou Guo, Zhihua Cai, and Shuhan Chen, 'Cubenet: X-shape connection for camouflaged object detection', *Pattern Recognition*, **127**, 108644, (2022).