https://doi.org/10.26599/AIR.2023.9150021 CAAI Artificial Intelligence Research

CamDiff: Camouflage Image Augmentation via Diffusion

Xue-Jing Luo[1], Shuo Wang[1, 2], Zongwei Wu[1], Christos Sakaridis[1], Yun Cheng[1], Deng-Ping Fan[1] ✉,
and Luc Van Gool[1]

ABSTRACT

The burgeoning field of Camouflaged Object Detection (COD) seeks to identify objects that blend into their surroundings. Despite the impressive performance of recent learning-based models, their robustness is limited, as existing methods may misclassify salient objects as camouflaged ones, despite these contradictory characteristics. This limitation may stem from the lack of multi-pattern training images, leading to reduced robustness against salient objects. To overcome the scarcity of multi-pattern training images, we introduce CamDiff, a novel approach inspired by AI-Generated Content (AIGC). Specifically, we leverage a latent diffusion model to synthesize salient objects in camouflaged scenes, while using the zero-shot image classification ability of the Contrastive Language-Image Pre-training (CLIP) model to prevent synthesis failures and ensure that the synthesized objects align with the input prompt. Consequently, the synthesized image retains its original camouflage label while incorporating salient objects, yielding camouflaged scenes with richer characteristics. The results of user studies show that the salient objects in our synthesized scenes attract the user's attention more; thus, such samples pose a greater challenge to the existing COD models. Our CamDiff enables flexible editing and effcient large-scale dataset generation at a low cost. It significantly enhances the training and testing phases of COD baselines, granting them robustness across diverse domains. Our newly generated datasets and source code are available at https://github.com/drlxj/CamDiff.

KEYWORDS

AI-generated content; diffusion model; camouflaged object detection; salient object detection

Camouflage is a predatory as well as defensive strategy that has evolved in natural objects through biological adaptation[1]. Visually, organisms alter the appearance of their bodies to match their surroundings, making them difficult to detect at first glance. Motivated by this phenomenon, a recent field of research called Camouflaged Object Detection (COD)[2–4] has gained significant attention from the computer vision community[5–7]. This area of study has broad applications, including medical image diagnosis and segmentation [8–10], species discovery[11], and crack inspection[12].

Several works[10, 13, 14] directly extend well-developed Salient Object Detection (SOD) towards COD tasks. Yet, salient and camouflaged objects are two contrasting object categories. The greater the level of saliency, the lower the degree of camouflage, and vice versa[15]. While the ideal case is to have a method that detects all objects, both salient and camouflaged, while noting the level of mimicry, misclassifying these two contrasting patterns with the same semantic label is not acceptable since it may hinder operational efficiency in various domains. For instance, misidentifying critical components or defects as camouflaged objects in manufacturing or quality control processes can lead to production errors, delays, or compromised product quality. In healthcare, misclassifying salient medical conditions, such as visible symptoms or anomalies, as camouflaged objects can result in misdiagnoses or delayed treatments, impacting patient outcomes, and hindering the effectiveness of medical interventions. Therefore, we argue that developing different strategies for detecting these two distinct object types is imperative. SOD models are based on global and local contrasts, whereas COD models should avoid such regions of high saliency. Unfortunately, our experiments reveal a decline in the accuracy of current COD methods when both salient and camouflaged objects co-exist in an image.

As Fig. 1 illustrates, we tested the robustness of several state-of-the-art COD methods, trained only with camouflaged samples, on salient objects. Many of these COD methods also detect objects when they are salient. These results indicate that the current COD models are not robust enough regarding scenes with salient objects. Specifically, the algorithms employed by PFNet[16] and ZoomNet[14], trained with COD datasets, detect only the more salient object (yellow ball) and neglect the less salient object (green ball). Thus, we speculate that existing COD works may only learn to distinguish the foreground and background rather than the camouflage and saliency patterns/prompts. This underscores the necessity to gain insight into camouflage patterns and make COD models effective.

To distinguish salient and camouflage patterns, one straightforward idea is to train the network via contrastive learning, which has demonstrated its effectiveness in other vision tasks[19–21]. As suggested in Refs. [22–24], strong data augmentation can significantly support contrastive learning, leading to effective feature representation modeling. However, generating positive and negative pairs as samples for contrastive training is not feasible in our setup due to the lack of salient objects in conventional camouflage datasets. Furthermore, existing COD datasets mainly contain a single object, making the direct

1 Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland
2 School of Systems Science, Beijing Normal University, Beijing 100875, China
Xue-Jing Luo and Shuo Wang contribute equally to this work.
Address correspondence to Deng-Ping Fan, dengpfan@gmail.com

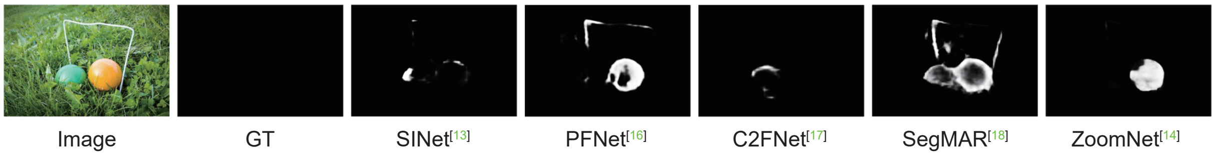

Fig. 1 Visual results of current COD models tested on an image with salient objects. As the object is salient, the ground truth (GT) should be all-black for the COD task. Nonetheless, the existing COD methods, especially PFNet and ZoomNet, are less robust to salient objects.

extension of contrastive learning infeasible. Besides, collecting and annotating a new dataset containing camouflaged and salient objects within a single image would be labor-intensive.

In this study, we aim to enhance the robustness of future COD models regarding salient objects. To achieve this objective, we propose augmenting contrastive samples in the training data by leveraging the recent diffusion model[25, 26] as a form of data augmentation to generate synthetic images. This approach is inspired by the success of AI-Generated Content (AIGC)[27, 28] and large-scale generative models. While some recent attempts have been made to utilize diffused images for data augmentation, these efforts are only feasible for more common scenarios such as daily indoor scenes[29] or urban landscapes[30] where the domain gap is small. By contrast, we are specifically interested in camouflage scenes, which are rare and challenging for pre-trained diffusion models. These differences make our task very challenging for synthesizing multi-pattern images with large domain gaps, which, to the best of our knowledge, has not been addressed in camouflage settings. In addition, existing works[31] rely on additional frozen-weight deep networks to generate pseudo-labels as supervision, limiting their performance and applications. These limitations motivate us to design a novel framework that generates realistic salient objects in the camouflage scenes. Our approach differs from the concurrent diffusion-augmentation methods[32, 33] regarding (a) the non-negligible domain gap, and (b) the preserved camouflage label.

To address the problem at hand, we propose a diffusion-based adversarial generation framework, named CamDiff. Specifically, our method consists of a generator and a discriminator. The generator is a frozen-weight Latent Diffusion Model (LDM)[25] that has been trained on a large number of categories, making it capable of synthesizing the most salient objects at scale. For the discriminator, we adopt the Contrastive Language-Image Pre-training (CLIP)[34] in an off-the-shelf manner for its generality. Our discriminator compares the input image prompt and the synthesized object to ensure semantic consistency. To preserve the original camouflage label, we only add the generated salient object in the background, i.e., outside the Ground-Truth (GT) label. Therefore, CamDiff effectively transforms the problem into an inpainting task, without requiring additional labeling cost. In this way, we can effectively and easily enable customized editing, hence improving the development of COD from the data-driven aspect.

Our main contributions are summarized as follows:

- We introduce CamDiff, which superimposes salient objects in camouflage scenes while preserving the original label. This framework facilitates collating and combining contrastive patterns within realistic images without incurring extra costs related to learning and labeling.
- We conduct experiments to test the robustness of the state-of-the-art COD methods on the COD test sets (i.e., Diff-COD), which are created from the original COD testing sets using CamDiff. Our results indicate that the current COD methods are not sufficiently robust against salient objects.
- To improve the resilience of current COD methods against salient objects, we generate a novel training set, called Diff-COD, from the original COD training sets using CamDiff. Our experimental results demonstrate that training the existing COD models on this new training set can enhance their robustness to salient objects.

Overall, our research provides a fresh perspective on the notion of “camouflage”, and our newly introduced camouflage synthesis tool will serve as a foundation for advancing this rapidly growing field.

1 Related Work

1.1 Diffusion models

Diffusion models[25, 26] are generative models that generate samples from a distribution by learning to remove noise from data points gradually. Recent research[35] shows that diffusion models outperform Generative Adversarial Networks (GANs)[36] in high-resolution image generation tasks without the drawbacks of mode collapse[37] and unstable training[38], and achieve unprecedented results in conditional image generation[28]. Therefore, they have been applied in many domains, such as text-to-image and guided synthesis[39, 40], 3D shape generation[41, 42], molecule prediction[43], video generation[44], and image inpainting[25].

Some researchers have studied the diffusion model for image inpainting. For example, Meng et al.[39] has found that diffusion models can not only fill regions of an image but can also accomplish it conditioned on a rough sketch of the image. Another study by Ref. [45] concludes that diffusion models can smoothly fill regions of an image with realistic content without edge artifacts when trained directly on the inpainting task.

1.2 Camouflage object detection

COD detects a concealed object within an image. Several research attention (e.g., SINet[13], UGTR[46], and ZoomNet[14]) have focused on the comparison of COD with SOD and concluded that simply extending SOD models to solve the COD task cannot bring the desired results because the target objects have different attributes, i.e., concealed or prominent. To detect the concealed image, many methods have been proposed recently. For example, some methods utilize a multi-stage strategy to solve the concealment of camouflaged images. SINet[13] is the first multi-stage method to locate and distinguish camouflaged objects. Another multi-stage method is SegMar[18], which localizes objects and zooms in on possible object regions to detect camouflaged objects progressively. In addition, the multi-scale feature aggregation is the second main strategy that has been used in many methods, such as CubeNet[47], which integrates low-level and high-level features by introducing X connection and attention fusion, as well as ZoomNet[14], which processes the input images at three different scales to fully explore imperceptible clues between the candidate objects and background surroundings. A detailed review of COD models is out of the scope of this work; we refer readers to recent top-tier works[4, 5].

In this paper, we focus on analyzing the robustness of end-to-end methods. Other generic models requiring additional post-processing are out of the scope of this paper. For example, the recent Segment Anything Model (SAM)[48] has shown great performance in generic segmentation tasks; extending such a method to COD tasks requires additional offline matching between all the candidate masks from SAM and the target GT box, as suggested in Ref. [49]. Therefore, our framework may not be directly beneficial for the vanilla SAM. However, a recent trend is to fine-tune the large-scale models on the downstream tasks with dedicated prompts. We believe that in such cases, our framework has great potential to improve prompt-aware SAM-variants.

1.3 Camouflage image generation

Although generating camouflage images has received limited attention, a few notable works exist in this area. One of the earliest methods, proposed in 2010, relies on hand-crafted features[1]. Zhang et al.[50] have recently proposed a deep learning-based approach for generating camouflaged images. Their method employs iterative optimization and attention-aware camouflage loss to selectively mask out salient features of foreground objects, while a saliency map ensures these features remain recognizable. However, the slow iterative optimization process limits the practical application of their method. Moreover, the style transfer of the background image to the hidden objects can often result in noticeable appearance discontinuities, leading to visually unnatural synthesized images. To overcome these limitations, Li et al.[51] has proposed a location-free camouflage generation network. Although this method outperforms the previous approach[50] in terms of visual quality, it may fail to preserve desired foreground features or make objects identifiable using the saliency map in certain cases. In summary, existing methods all follow the same strategy to produce camouflage images: they use two images to represent the foreground image and the background image, respectively, and then attempt to directly integrate the foreground image with the background image by finding a place where the foreground object is hard to detect within the synthesized image. Notably, most of these methods only synthesize new COD images without providing the associated masks. Therefore, additional labeling is always required for supervised learning. Differently, we maintain the camouflaged ground-truth masks and blend the salient objects into the background with the help of a trained diffusion model, allowing us to benefit from the original COD masks while enriching the scene with more patterns.

2 Proposed CamDiff

2.1 Overall architecture

To evaluate the effectiveness of existing COD methods on negative samples (i.e., scenes with salient objects), we suggest creating synthetic salient objects on top of current camouflage datasets. Normally, when a task-specific model is trained with COD datasets, it should effectively detect the camouflaged samples, while being robust and not detecting the synthesized salient objects. Therefore, such an approach allows us to thoroughly investigate whether a learning-based COD method can accurately distinguish between camouflage and salient objects.

To achieve this objective, we propose a new generation network called CamDiff, which is built upon existing COD datasets. Since these datasets already contain camouflaged objects with corresponding camouflage ground truth masks, our aim is to add synthesized salient objects into the background. By doing so, we can maintain the original camouflage labels and leverage them while also introducing salient samples that have contrasting characteristics.

Figure 2 illustrates the overall architecture of our proposed method. We start with a COD dataset, which provides us with a source image and its corresponding GT. Using the GT, we identify the bounding box with the minimum coverage area to prevent CamDiff from altering the camouflaged image. Next, we divide the source image into nine areas via grid lines, using the bounding box to preserve the area where the camouflaged object is placed. Only eight of the areas are available for input into CamDiff. We randomly select one of these regions and cut it out from the source image, covering a specific proportion (e.g., 75% as the default setting in our experiments) of the total area from the center. We then feed the masked image into the generation network, and CamDiff generates a salient object within the masked area. Finally, we place the selected region back into its original location within the source image. In such a manner, we can not only preserve the GT labels for camouflaged objects but also add contradictory synthesized salient samples.

To generate the salient object, we propose a generation framework based on the GAN architecture. Specifically, we utilize the widely-acknowledged LDM as the generator and the CLIP as the discriminator. As shown in Fig. 2, the input to our framework is an image with the previously-masked region, along with a text prompt that describes the target object. This masked region and text prompt are then fed into the generator. Based on the prompt, the LDM block generates the target object on top of the masked region. The filled-up region is then sent to the discriminator to determine if it matches the input prompt. If not, the generator adjusts the seed to generate a new salient object. The objective is to train the generation network to only produce validated images when the discriminator predicts a high probability of matching the input prompt.

Our framework transforms the image generation task into an inpainting task, and thus requires a mask to cover the selected region. The mask generation process is explained in Algorithm 1.

The mask is designed to cover a certain percentage of the selected region to avoid artifacts when blending the synthesized object with the source image. The ratio of the masked area to the region area is set to a constant, $\text{RATIO}_{\text{MASK}}$. The size of the selected region is crucial for the inpainting task, as it can affect the quality of the generated salient object. If the region is too small, the LDM may fill the background instead of the object, while if it is too large, the salient object may be too much larger than the concealed object, misleading COD methods. Therefore, we set an upper bound ($\text{RATIO}_{\text{MAX}}$) and a lower bound ($\text{RATIO}_{\text{MIN}}$) for the ratio between the region area and the total area of the source image. The values for these parameters are listed in Table. 1.

2.2 Latent diffusion model

We use the LDM[25] which is pre-trained on a large-scale dataset as our generator's base model. The LDM is a two-stage method that consists of an autoencoding model to learn the latent representation of an image and a Denoising Diffusion Probabilistic Model (DDPM)[26]. In the first stage, the autoencoding model is trained to learn a space that is perceptually equivalent to the image space. The encoder $\mathscr{E}$ encodes the given image $x \in \mathbb{R}^{H\times W\times 3}$ to the latent representation $z \in \mathbb{R}^{H\times W\times C}$ so that $z = \mathscr{E}(x)$, while the deocder $\mathscr{D}$ reconstructs the estimated image $\tilde{x}$ from the latent representation, such that $\tilde{x} = \mathscr{D}(\tilde{z})$ and $\tilde{x} \approx x$. In the second stage, the DDPM is trained to generate the latent representation within the pre-trained

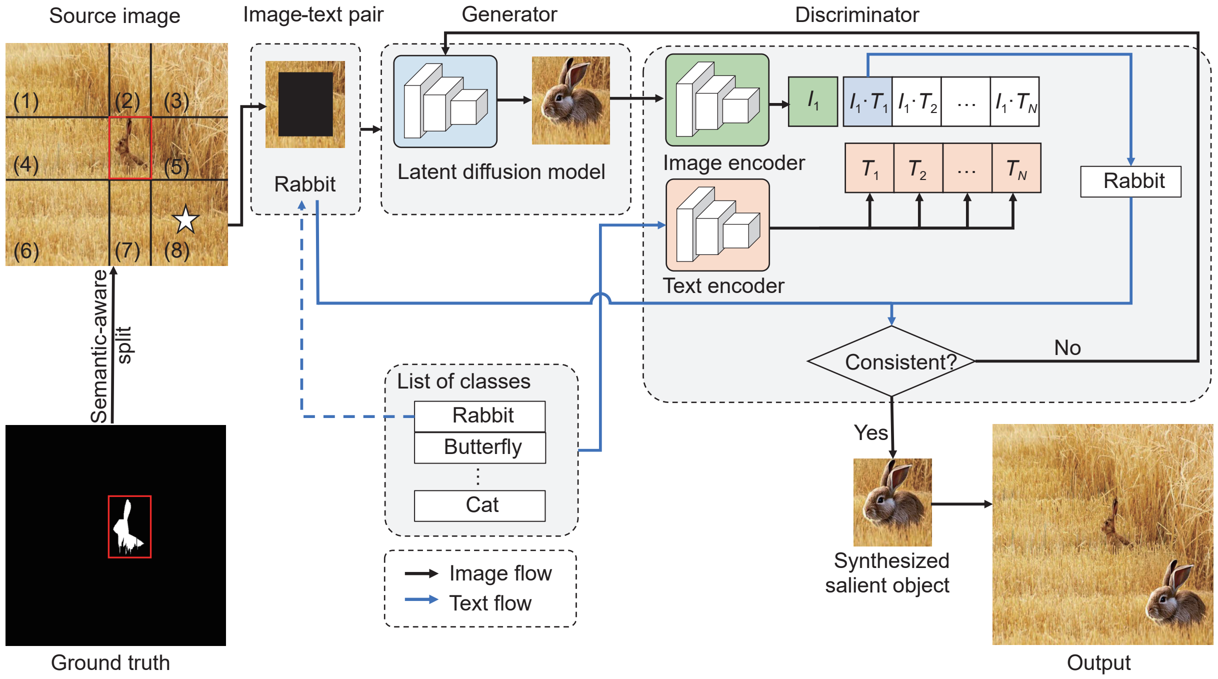

Fig. 2 Our CamDiff consists of a generator and a discriminator. The input of CamDiff is a pair of a masked image and a text prompt. Only after the discriminator judges that the synthesized object is consistent with the text input, the synthesized image can be output and placed back into the source image. The white star in the source image means that region (8) is selected as the masked region.

Algorithm 1 Mask generation.

```
Put the eight regions' index in a list candidates in order
Shuffle the index in candidates
for $i$ in candidates do
    if the area of region $i$ is higher than $\mathrm{RATIO_{MIN}}$ then
        if the area of region $i$ is less than $\mathrm{RATIO_{MAX}}$ then
            choose the area mask that covers $\mathrm{RATIO_{MASK}}$ of the total region area from the center
            break
        else
            choose the area mask that covers $\mathrm{RATIO_{MASK}} \cdot \mathrm{RATIO_{MAX}}$ of the total region area from the center
            break
        end if
    else
        continue
    end if
end for
return mask
```

Table 1 Hyperparameters setting.

Parameter	Value
$\mathrm{RATIO_{MIN}}$	6.25%
$\mathrm{RATIO_{MAX}}$	25%
$\mathrm{RATIO_{MASK}}$	75%

latent space based on a random Gaussian noise input z_t. The neural backbone $\varepsilon_\theta(z_t, t)$ of the LDM is realized as a time-conditional UNet, and the objective of the DDPM trained on latent space is simplified as:

$$L_{\mathrm{DM}} := \boldsymbol{E}_{\mathscr{E}(x),\varepsilon\sim\mathcal{N}(0,1),t}\left[\|\varepsilon - \varepsilon_\theta(z_t, t)\|_2^2\right] \tag{1}$$

2.3 Conditioning LDM

To control the image synthesis, the conditional LDM implements a conditional denoising autoencoder $\varepsilon_\theta(z_t, y, t)$ through inputs y such as text, semantic maps, or other image-to-image translation tasks[25]. The proposed CamDiff exploits this ability to control image synthesis through text input. To turn DDPMs into more flexible conditional image generators, their underlying UNet backbone is augmented with the cross-attention mechanism. The embedding sequences $\tau_\theta(y) \in \mathbb{R}^{M\times d_\tau}$ from the CLIP ViT-L/14 encoder is fused with latent feature maps via a cross-attention layer implementing as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}} \cdot V\right) \tag{2}$$

where $Q = W_Q^{(i)} \cdot \phi_i(z_t)$, $K = W_K^{(i)} \cdot \tau_\theta(y)$, $V = W_V^{(i)} \cdot \tau_\theta(y)$, and $\phi_i(z_t)$ is a intermediate representation of the UNet implementing ε_θ. $W_Q^{(i)}$, $W_K^{(i)}$, and $W_V^{(i)}$ are learnable projection matrix. The objective of the conditional LDM is converted from Eq. (1) to

$$L_{\mathrm{CDM}} := \boldsymbol{E}_{\mathscr{E}(x),y,\varepsilon\sim\mathcal{N}(0,1),t}\left[\|\varepsilon - \varepsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2\right] \tag{3}$$

2.4 CLIP for zero-shot image classification

To improve the quality of generated objects based on text input, it is necessary to use a discriminator that can assess the consistency of the generated objects with the text prompt. However, since the text prompt can be any arbitrary class, traditional classifiers that only recognize a fixed set of object categories are unsuitable for this task. Therefore, CLIP models offer a better option for this task.

The CLIP model comprises an image encoder and a text encoder. The image encoder can employ various computer vision architectures, including five ResNets of varying sizes and three vision transformer architectures. Meanwhile, the text encoder is a decoder-only transformer that uses masked self-attention to ensure that the transformer's representation for each token in a sequence depends solely on tokens that appear before it. This approach prevents any token from looking ahead to inform its representation better. Both encoders undergo pre-training to align similar text and images in vector space. This is achieved by taking image-text pairs and pushing their output vectors closer in vector space while separating the vectors of non-pairs. The CLIP model is trained on a massive dataset of 400 million text-image pairs publicly available on the internet. Since the image and text encoders of the CLIP model have already been trained on diverse, unfiltered, and noisy data, in our application, we freeze CLIP parameters to benefit its generalization capability, enabling our approach to be performed in a zero-shot manner.

In CamDiff, the image with the synthesized object is encoded via the image encoder, and the text encoder encodes the list of all classes. As shown in Fig. 2, the embedding which is output by the image encoder is combined via a dot-product operation with the embedding of each class generated by the text encoder. The highest value of the resulting output vector among all classes represents the class with the embedding which is most consistent with the image. If the class with the highest value is consistent with the language prompt given by the user, then the synthesized image can be placed back into the original image and subsequently output.

3 Experiment

3.1 Experimental Setup

Datasets. To synthesize multi-pattern images for the COD task, we selected four widely-used COD datasets: CAMO[52], CHAM[53], COD10K[13], and NC4K[54].

It should be noted that the COD10K dataset provides semantic labels as filenames. Therefore, we used the label directly as the text prompt. Some prompts are shown in Fig. 2, which lists the classes. However, the list of classes is not directly available for the other three datasets. Since they contain common animal species such as birds, cats, dogs, etc., we randomly chose a text prompt from the COD10K label list.

Baselines. To evaluate the robustness of existing COD methods to both salient and camouflaged objects, we selected four representative and classical COD methods: SINet[13], PFNet[16], C2FNet[17], and ZoomNet[14], as our baselines. It is worth noting that since our paper submission, several new state-of-the-art models have emerged, including FSPNet[5] and EVP model[55]. However, this paper aims to explore new mechanisms for detecting camouflage patterns, and thus comprehensive testing of all models falls beyond the scope of this article.

Evaluation metrics. To assess the quality of the synthesized image, we employed Inception scores[56]. A higher Inception score in the context of image generation models means that the generated images are of better quality and more diverse. For COD models, we evaluated the performance using 4 golden metrics: mean absolute error (M), max F-measure (F_m), S-measure (S_m), and max E-measure (E_m).

Implementation details. Our implementation of CamDiff is realized in the Pytorch framework, with hyperparameters related to mask generation specified in Table 1. The whole learning process is executed on a 2080Ti GPU. We followed the conventional train-test split[2, 13, 14, 47], using a training set of 4040 images from COD10K and CAMO.

Among these training samples, we replaced 3717 images with our synthesized multi-pattern images. The original testing samples comprised 6473 images from CAMO, CHAM, COD10K, and NC4K. To form our Diff-COD testing set, we replaced 5395 images with our generated images. Although we cannot entirely replace the camouflage dataset since some images contain specific objects that the diffusion model may not generate well using the pre-trained weights, our success rate remains high. Specifically, over 92% of the training images and 83% of the testing images can be modified with extra salient patterns. This high success rate confirms the effectiveness of our generation framework. Note that we resized the images and masks to 512×512 to meet the requirements of the LDM.

3.2 Quality of synthesized images

Inception score. To prove that our CamDiff can generate a prominent object rather than a concealed object, we choose the inception score as the evaluation metric and evaluate it on the SOD datasets[57–59], COD datasets[13, 52–54], and our generated dataset with multi-pattern images. Table 2 shows that the original SOD datasets have a higher inception score than the original COD dataset, which aligns with our expectations. The rationale behind the Inception score is that a well-synthesized image should contain easily recognizable objects for an off-the-shelf recognition system. The recognition system is more likely to detect prominent objects than camouflage ones. As a result, images with multi-patterns tend to have a higher Inception score than those with camouflaged patterns.

By comparing the Inception score before and after the modification, we can easily evaluate the effectiveness of our framework. Replacing images in the COD dataset with multi-pattern images shows that the inception score has increased across all COD datasets. This indicates that we have successfully incorporated prominent patterns on top of the original COD datasets.

User study. We also conducted a user study to evaluate the synthesized images' quality. The objective is to find the target object from images depending upon the prompt (e.g., "Butterfly" in Fig. 3).

Participants were given a subset of our synthesized images and were asked to circle the object they detected first based on the corresponding labels. The salient object chosen by the user is considered the most prominent since it attracts the most human attention. The results of our user study, with over 10 participants, show that the average rate of users choosing the synthesized

Table 2 Comparision of the generated dataset with the original COD and SOD dataset. The type "Orig." means the original dataset, while the type "New" means the synthesized dataset based on the corresponding COD dataset.

Dataset		Type	Inception score ↑
SOD	DUTSE-TE	Orig.	71.63
	ECSSD	Orig.	24.40
	XPIE (salient)	Orig.	96.79
	XPIE (not salient)	Orig.	13.96
COD	CAMO	Orig.	6.61
		New	9.90
	CHAM	Orig.	4.38
		new	5.98
	COD10K	Orig.	7.00
		New	14.85
	NC4K	Orig.	7.00
		New	12.87

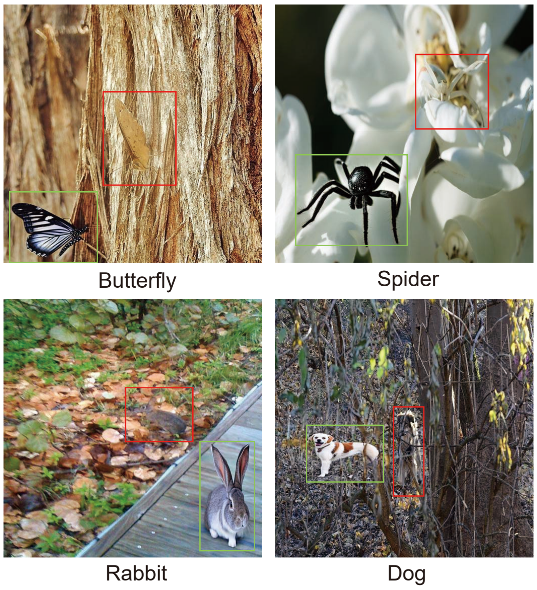

Fig. 3 In the user study, the solution involved presenting the synthesized object within a green box, while the original object within the image is enclosed in a red box. The study results indicate that users are likelier to circle the objects in the green box, highlighting the synthesized objects as more prominent and easier to detect than the original objects within the images.

object, i.e., the salient ones, is 98%. This indicates that the synthesized objects are more prominent and easier to detect than the original objects in the images.

Overall, the increased inception score and positive results from the user study support our claim that CamDiff generates prominent objects rather than concealed ones in the synthesized images. In addition, CamDiff has demonstrated its robust capability to generate diverse objects and variations in posture for a single object type. Figure 4 provides examples of various classes of synthesized images, each of which can be extended to generate three additional images of the same class.

3.3 Quantitative comparison

We introduce quantitative experiments and evaluate state-of-the-art COD methods on the synthesized samples generated by CamDiff. Table 3 shows the performance of pre-trained models on original and generated testing samples; Table 4 compares the performance trained with original COD images and our generated training samples; Table 5 presents the robustness analysis on SOD datasets.

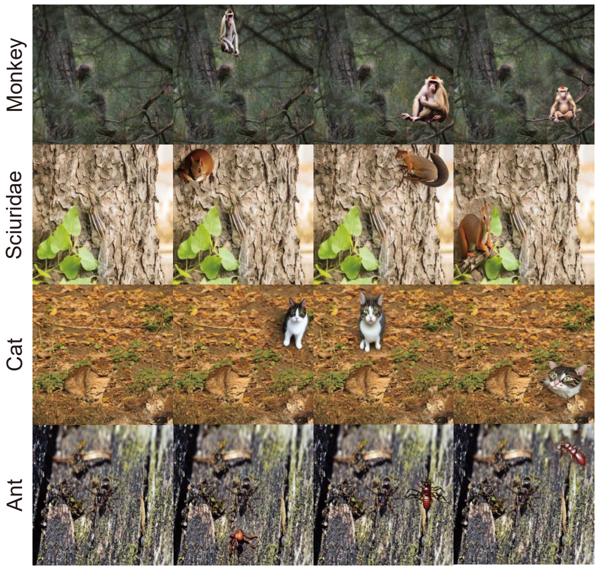

Fig. 4 Examples of the synthesized images from CamDiff from various classes. Each image is extended to generate three additional images of the same class, featuring objects with varying appearances.

Pre-trained weights. We created a new Diff-COD dataset to evaluate the effectiveness of existing COD methods on images containing salient and camouflaged objects. This dataset includes both types of images, and we trained 4 state-of-the-art COD models (SINet[13], PFNet[16], C2FNet[17], and ZoomNet[14]) on the Diff-COD training set.

We then evaluated their performance on the Diff-COD testing set. It is important to note that the pre-trained LDM module block can only output images with a resolution of 512×512. This resolution is suitable for most existing methods trained with a resolution less than 352×352.

Table 3 Quantitative results of the pre-trained COD models on Diff-COD test dataset and COD dataset. ↑ (↓) denotes that the higher (lower) is better.

Dataset		SINet[13]	PFNet[16]	C2FNet[17]	ZoomNet[14]
CAMO	$M\downarrow$	0.099	0.085	0.079	0.066
	$F_m\uparrow$	0.762	0.793	0.802	0.832
	$S_m\uparrow$	0.751	0.782	0.796	0.819
	$E_m\uparrow$	0.790	0.845	0.856	0.881
Diff-CAMO	$M\downarrow$	0.130	0.122	0.116	0.136
	$F_m\uparrow$	0.581	0.626	0.632	0.557
	$S_m\uparrow$	0.651	0.686	0.700	0.664
	$E_m\uparrow$	0.768	0.792	0.802	0.790
CHAM	$M\downarrow$	0.044	0.033	0.032	0.023
	$F_m\uparrow$	0.845	0.859	0.871	0.883
	$S_m\uparrow$	0.868	0.882	0.888	0.900
	$E_m\uparrow$	0.908	0.927	0.936	0.944
Diff-CHAM	$M\downarrow$	0.065	0.065	0.061	0.088
	$F_m\uparrow$	0.700	0.795	0.726	0.596
	$S_m\uparrow$	0.787	0.708	0.798	0.726
	$E_m\uparrow$	0.869	0.865	0.869	0.850
COD10K	$M\downarrow$	0.051	0.040	0.036	0.029
	$F_m\uparrow$	0.708	0.747	0.764	0.799
	$S_m\uparrow$	0.771	0.800	0.813	0.836
	$E_m\uparrow$	0.832	0.880	0.894	0.887
Diff-COD10K	$M\downarrow$	0.057	0.054	0.052	0.064
	$F_m\uparrow$	0.620	0.644	0.656	0.585
	$S_m\uparrow$	0.727	0.751	0.757	0.729
	$E_m\uparrow$	0.826	0.832	0.839	0.841
NC4K	$M\downarrow$	0.058	0.053	0.049	0.044
	$F_m\uparrow$	0.804	0.820	0.831	0.845
	$S_m\uparrow$	0.808	0.829	0.838	0.851
	$E_m\uparrow$	0.873	0.891	0.898	0.896
Diff-NC4K	$M\downarrow$	0.090	0.084	0.080	0.076
	$F_m\uparrow$	0.640	0.664	0.666	0.631
	$S_m\uparrow$	0.719	0.744	0.746	0.739
	$E_m\uparrow$	0.821	0.830	0.834	0.841

Table 4 Quantitative results of the test Diff-COD dataset. "Pre." means the model is loaded with the pre-trained checkpoint provided by the officially released code. "Tr." means that the model is loaded by the checkpoints trained on our synthesized training set.

Dataset		SINet[13]		PFNet[16]		C2FNet[17]		ZoomNet[14]	
		Pre.	Tr.	Pre.	Tr.	Pre.	Tr.	Pre.	Tr.
Diff-CAMO	$M\downarrow$	0.130	0.094	0.122	0.087	0.116	0.078	0.136	0.092
	$F_m\uparrow$	0.581	0.769	0.626	0.787	0.632	0.800	0.557	0.758
	$S_m\uparrow$	0.651	0.753	0.686	0.773	0.700	0.789	0.664	0.773
	$E_m\uparrow$	0.768	0.802	0.792	0.828	0.802	0.848	0.790	0.803
Diff-CHAM	$M\downarrow$	0.065	0.036	0.065	0.033	0.061	0.030	0.088	0.058
	$F_m\uparrow$	0.700	0.864	0.795	0.858	0.726	0.870	0.596	0.764
	$S_m\uparrow$	0.787	0.884	0.708	0.880	0.798	0.888	0.726	0.816
	$E_m\uparrow$	0.869	0.931	0.865	0.933	0.869	0.949	0.850	0.845
Diff-COD10K	$M\downarrow$	0.057	0.047	0.054	0.041	0.052	0.038	0.064	0.053
	$F_m\uparrow$	0.620	0.708	0.644	0.735	0.656	0.748	0.585	0.691
	$S_m\uparrow$	0.727	0.773	0.751	0.794	0.757	0.801	0.729	0.770
	$E_m\uparrow$	0.826	0.849	0.832	0.874	0.839	0.887	0.841	0.805
Diff-NC4K	$M\downarrow$	0.090	0.060	0.084	0.052	0.080	0.047	0.076	0.069
	$F_m\uparrow$	0.640	0.807	0.664	0.821	0.666	0.834	0.631	0.789
	$S_m\uparrow$	0.719	0.811	0.744	0.830	0.746	0.840	0.739	0.814
	$E_m\uparrow$	0.821	0.866	0.830	0.894	0.834	0.905	0.841	0.847

Table 5 Quantitative results of the original SOD testing sets. "Pre." means the model is loaded with the pre-trained checkpoint provided by the paper, while "Tr." means that the model is loaded by the checkpoints trained on our synthesized training set.

Dataset		SINet[13]		PFNet[16]		C2FNet[17]		ZoomNet[14]	
		Pre.	Tr.	Pre.	Tr.	Pre.	Tr.	Pre.	Tr.
DUTS-TE	$M\downarrow$	0.065	0.082	0.064	0.079	0.065	0.069	0.080	0.083
	$F_m\uparrow$	0.820	0.760	0.808	0.748	0.807	0.780	0.715	0.718
	$S_m\uparrow$	0.806	0.741	0.806	0.751	0.802	0.777	0.772	0.768
	$E_m\uparrow$	0.846	0.757	0.845	0.778	0.832	0.812	0.840	0.842
ECSSD	$M\downarrow$	0.106	0.135	0.105	0.130	0.116	0.115	0.129	0.134
	$F_m\uparrow$	0.844	0.784	0.822	0.762	0.802	0.790	0.744	0.751
	$S_m\uparrow$	0.766	0.692	0.766	0.703	0.748	0.734	0.722	0.715
	$E_m\uparrow$	0.786	0.688	0.784	0.702	0.750	0.740	0.834	0.841
XPIE-SAL	$M\downarrow$	0.090	0.119	0.093	0.115	0.099	0.101	0.115	0.123
	$F_m\uparrow$	0.822	0.763	0.804	0.739	0.786	0.762	0.720	0.703
	$S_m\uparrow$	0.770	0.691	0.762	697	0.749	0.728	0.723	0.705
	$E_m\uparrow$	0.805	0.697	0.792	0.709	0.768	0.749	0.820	0.815

However, the current state-of-the-art method, ZoomNet[5], requires a main resolution of 384×384 and an additional higher resolution with a scale of 1.5 (576×576), which is larger than the capacity of the LDM model. To ensure a fair comparison, we retrained ZoomNet with a main scale of 288×288. To ensure equal evaluation, we trained ZoomNet on the original and our new training sets with the same main resolution of 288×288.

Table 3 compares each model's performance with its pre-trained checkpoints on both Diff-COD and original COD datasets. The results indicate that all COD methods perform significantly worse on the Diff-COD dataset. This is because these methods detect the additionally generated salient object and classify them as camouflage ones, indicating a lack of robustness to saliency. As a result, we can conclude that our Diff-COD testing set serves as a more challenging benchmark and can be used as an additional tool for robustness analysis.

Trained on Our Generated Datasets. As previously mentioned, our framework has the capability to generate new training samples with both salient and camouflage objects. By training on our Diff-COD dataset using only camouflage supervision, the networks should learn the distinction between the two contrasting notions and become more resilient to saliency.

Table 4 displays the results of the pre-trained COD models trained with original COD training sets and the newly-trained COD models on our Diff-COD training sets. It is evident that the models trained on the Diff-COD training set perform significantly better on the Diff-COD testing set compared to their counterparts.

To further confirm the effectiveness of our approach in enhancing the robustness of COD models against saliency, we conducted experiments on conventional saliency datasets, including DUTS-TE[57], ECSSD[58], XPIE[59]. As displayed in Table 5, when the models were trained using our Diff-COD dataset, their performance on saliency benchmarks declined. This is expected since the poorer performance on the SOD datasets indicates that the newly-trained models have truly learned the camouflage pattern but not the salient pattern. As a result, these models are better equipped to withstand the influence of salient objects.

3.4 Qualitative comparison

Figure 5 demonstrates the effect of training on multi-pattern images on the performance of COD models. The figure is divided into three cases, each presenting the results for a different camouflaged object (fish, crab, and frog). On the left side of the dashed line in each case, the original image from the COD dataset, a synthesized multi-pattern image, and the ground truth are

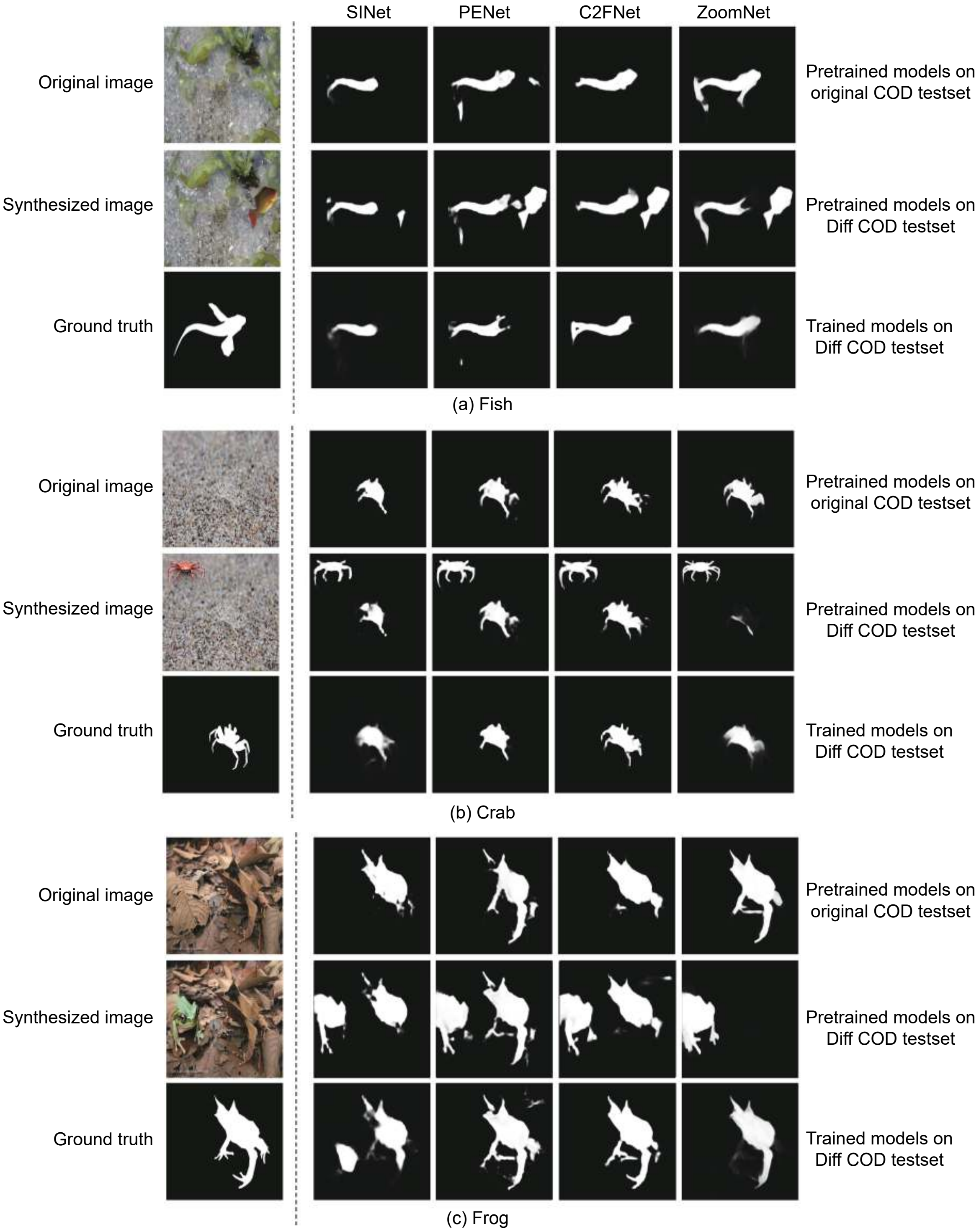
SINet
PENet
C2FNet
ZoomNet
Original image
Synthesized image
Ground truth
Pretrained models on original COD testset
Pretrained models on Diff COD testset
Trained models on Diff COD testset
(a) Fish
Original image
Synthesized image
Ground truth
Pretrained models on original COD testset
Pretrained models on Diff COD testset
Trained models on Diff COD testset
(b) Crab
Original image
Synthesized image
Ground truth
Pretrained models on original COD testset
Pretrained models on Diff COD testset
Trained models on Diff COD testset
(c) Frog

Fig. 5 Qualitative Comparison. We conducted a qualitative comparison on three cases: Fish, Crab, and Frog. We analyzed the impact of adding salient objects to camouflaged images on pre-trained SINet, PFNet, C2FNet, and ZoomNet, respectively, by comparing the results of the first two rows. Furthermore, we evaluated the training results on the Diff-COD test set by comparing the qualitative outcomes with the pre-trained results.

shown. The right side displays the results of four pre-trained models (SINet, PFNet, C2FNet, and ZoomNet) on the original COD datasets in the first row. The second row of the illustration presents the results of the models tested on the synthesized images using the same checkpoints as in the first row. Most of them detect salient objects, which is undesirable, and the accuracy of detecting camouflaged objects decreases. For instance, SINet loses some parts compared with the mask in the first row, and ZoomNet ignores camouflaged objects. These results indicate that COD methods lack robustness to saliency. The third row of the illustration presents the results of the models trained on our Diff-COD dataset and then tested on the synthesized images. Compared to the second row, the robustness to saliency improves significantly. Nevertheless, compared to the first row, ZoomNet loses some parts of the camouflaged object. We believe this may be caused by adding noise in the training set making the fitting more difficult, but we plan to evaluate the cause in future work.

Overall, it can be concluded from Fig. 5 that the presence of salient objects harms the performance of COD models in detecting camouflaged objects. However, training the COD models on multi-pattern images increases their robustness to the effects of salient objects.

4 Conclusion

In summary, our work introduces CamDiff, a framework that generates salient objects while preserving the original label on camouflage scenes, enabling the easier collation and combination of contrastive patterns in realistic images without incurring extra costs related to learning and labeling. Through experiments conducted on Diff-COD test sets, we demonstrate that current COD methods lack robustness to negative examples (e.g., scenes with salient objects). To address this limitation, we create a novel Diff-COD training set using CamDiff. By generating multi-pattern images with both salient and camouflaged objects, CamDiff provides a more challenging and representative dataset for training COD models, leading to better performance in real-world scenarios where camouflaged objects may be more difficult to detect due to the presence of salient objects. In this way, we hope that future COD methods have the potential to improve the performance in COD by distinguishing between salient objects and camouflaged ones. Our experimental results demonstrate that training existing COD models on this set improves their resilience to salient objects. Overall, our work provides a new perspective on camouflage and contributes to the development of this emerging field.

Future work. We aim to extend our framework to consider original images with multiple objects and save room for their generation. Additionally, while we only implemented multi-pattern images as the data augmentation method in our experiments, we plan to evaluate the results using other data augmentation methods to provide a more comprehensive analysis of the impact of multi-pattern images on the performance and robustness of these models.

Article History

Received: 12 April 2023; Revised: 23 May 2023; Accepted: 6 October 2023

References

[1] H. K. Chu, W. H. Hsu, N. J. Mitra, D. Cohen-Or, T. T. Wong, and T. Y. Lee, Camouflage images, *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–8, 2010.

[2] D. P. Fan, G. P. Ji, M. M. Cheng, and L. Shao, Concealed object detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, 2022.

[3] R. He, Q. Dong, J. Lin, and R. W. H. Lau, Weakly-supervised camouflaged object detection with scribble annotations, *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 1, pp. 781–789, 2023.

[4] X. Hu, S. Wang, X. Qin, H. Dai, W. Ren, D. Luo, Y. Tai, and L. Shao, High-resolution iterative feedback network for camouflaged object detection, *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 1, pp. 881–889, 2023.

[5] Z. Huang, H. Dai, T. Z. Xiang, S. Wang, H. X. Chen, J. Qin, and H. Xiong, Feature shrinkage pyramid for camouflaged object detection with transformers, in *Proc. 2023 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, BC, Canada, 2023, pp. 5557–5566.

[6] Z. Wu, D. P. Paudel, D. -P. Fan, J. Wang, S. Wang, C. Demonceaux, R. Timofte, and L. Van Gool, Source-free depth for object pop-out, arXiv preprint arXiv: 2212.05370, 2022.

[7] Y. Zhong, B. Li, L. Tang, S. Kuang, S. Wu, and S. Ding, Detecting camouflaged object in frequency domain, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 4494–4503.

[8] H. Ali Qadir, Y. Shin, J. Solhusvik, J. Bergsland, L. Aabakken, and I. Balasingham, Polyp detection and segmentation using mask R-CNN: Does a deeper feature extractor CNN always perform better? in *Proc. 2019 13th Int. Symp. on Medical Information and Communication Technology* (*ISMICT*), Oslo, Norway, 2019, pp. 1–6.

[9] F. Ucar and D. Korkmaz, COVIDiagnosis-Net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images, *Med. Hypotheses*, vol. 140, pp. 109761, 2020.

[10] B. Dong, W. Wang, D. P. Fan, J. Li, H. Fu, and L. Shao, Polyp-PVT: Polyp segmentation with pyramid vision transformers, arXiv preprint arXiv: 2108.06932, 2021.

[11] R. Pérez-de la Fuente, X. Delclòs, E. Peñalver, M. Speranza, J. Wierzchos, C. Ascaso, and M. S. Engel, Early evolution and ecology of camouflage in insects, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 109, no. 52, pp. 21414–21419, 2012.

[12] F. Fang, L. Li, Y. Gu, H. Zhu, and J. H. Lim, A novel hybrid approach for crack detection, *Pattern Recognit.*, vol. 107, pp. 107474, 2020.

[13] D. P. Fan, G. P. Ji, G. Sun, M. M. Cheng, J. Shen, and L. Shao, Camouflaged object detection, in *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Seattle, WA, USA, 2020, pp. 2774–2784.

[14] Y. Pang, X. Zhao, T. Z. Xiang, L. Zhang, and H. Lu, Zoom in and out: A mixed-scale triplet network for camouflaged object detection, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 2150–2160.

[15] A. Li, J. Zhang, Y. Lv, B. Liu, T. Zhang, and Y. Dai, Uncertainty-aware joint salient object and camouflaged object detection, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 10066–10076.

[16] H. Mei, G. P. Ji, Z. Wei, X. Yang, X. Wei, and D. P. Fan, Camouflaged object segmentation with distraction mining, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 8768–8777.

[17] Y. Sun, G. Chen, T. Zhou, Y. Zhang, and N. Liu, Context-aware cross-level fusion network for camouflaged object detection, in *Proc. 30th Int. Joint Conf. Artificial Intelligence* (*IJCAI-21*), virtual, 2021, pp. 1025–1031.

[18] Q. Jia, S. Yao, Y. Liu, X. Fan, R. Liu, and Z. Luo, Segment, magnify and reiterate: Detecting camouflaged objects the hard way, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 4703–4712.

[19] B. Dai and D. Lin, Contrastive learning for image captioning, in *Proc. 31st Conf. Neural Information Processing Systems* (*NIPS 2017*), Long Beach, CA, USA, pp. 898–907.

[20] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. Langlotz, Contrastive learning of medical visual representations from paired images and text, in *Proc. 9th Int. Conf. Learning Representations*, virtual, 2021.

[21] M. Kang and J. Park, ContraGAN: contrastive learning for conditional image generation, in *Proc. 34th Conf. Neural Information Processing Systems* (*NeurIPS 2020*), Vancouver, Canada, 2020.

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, A simple framework for contrastive learning of visual representations, arXiv preprint arXiv: 2002.05709, 2020.

[23] Yonglong Tian, Chen Sun, B. Poole, D. Krishnan, and P. Isola, What makes for good views for contrastive learning? in *Proc. 34th Conf. Neural Information Processing Systems* (*NeurIPS 2020*), Vancouver, Canada, 2020.

[24] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, Supervised contrastive learning, in *Proc. 34th Conf. Neural Information Processing Systems* (*NeurIPS 2020*), Vancouver, Canada, 2020.

[25] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 10674–10685.

[26] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, in *Proc. 34th Conf. Neural Information Processing Systems* (*NeurIPS 2020*), Vancouver, Canada, 2020.

[27] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, A comprehensive survey of AI-generated content (AIGC): A history of generative AI from GAN to ChatGPT, arXiv preprint arXiv: 2303.04226, 2023.

[28] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, Hierarchical text-conditional image generation with CLIP latents, arXiv preprint arXiv: 2204.06125, 2022.

[29] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S. C. Zhu, Diffusion-based generation, optimization, and planning in 3D scenes, arXiv preprint arXiv: 2301.06015, 2023.

[30] C. H. Lin, H. Y. Lee, W. Menapace, M. Chai, A. Siarohin, M. H. Yang, and S. Tulyakov, InfiniCity: infinite-scale city synthesis, arXiv preprint arXiv: 2301.09637, 2023.

[31] D. Huynh, J. Kuen, Z. Lin, J. Gu, and E. Elhamifar, Open-vocabulary instance segmentation via robust cross-modal pseudo-labeling, in *Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), New Orleans, LA, USA, 2022, pp. 7010–7021.

[32] Y. Benigmim, S. Roy, S. Essid, V. Kalogeiton, and S. Lathuilière, One-shot unsupervised domain adaptation with personalized diffusion models, arXiv preprint arXiv: 2303.18080, 2023.

[33] M. Zhang, X. Guo, L. Pan, Z. Cai, F. Hong, H. Li, L. Yang, and Z. Liu, ReMoDiffuse: retrieval-augmented motion diffusion model, arXiv preprint arXiv: 2304.01116, 2023.

[34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, and I. Sutskever, Learning transferable visual models from natural language supervision, in *Proc. 38 th Int. Conf. Machine Learning*, virtual, 2021, pp. 8748–8763.

[35] P. Dhariwal and A. Nichol, Diffusion models beat GANs on image synthesis, arXiv preprint arXiv: 2105.05233, 2021.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial networks, *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[37] C. Nash, J. Menick, S. Dieleman, and P. W. Battaglia, Generating images with sparse representations, in *Proc. 38 th Int. Conf. Machine Learning*, virtual, 2021, pp. 7958–7968.

[38] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, Spectral normalization for generative adversarial networks, in *Proc. 6th Int. Conf. Learning Representations* (*ICLR*), Vancouver, Canada, 2018.

[39] C. Meng, Y. Song, J. Song, J. Wu, and S. Ermon, SDEdit: Image synthesis and editing with stochastic differential equations, in *Proc. 10th Int. Conf. Learning Representations* (*ICLR*), virtual, 2022.

[40] B. Poole, A. Jain, J. Barron, and B. Mildenhall, DreamFusion: Text-to-3D using 2D diffusion, in *Proc. 11th Int. Conf. Learning Representations* (*ICLR*), Kigali, Rwanda, 2023.

[41] L. Zhou, Y. Du, and J. Wu, 3D shape generation and completion through point-voxel diffusion, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Montreal, Canada, 2022, pp. 5806–5815.

[42] S. Luo and W. Hu, Diffusion probabilistic models for 3D point cloud generation, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 2836–2844.

[43] B. L. Trippe, J. Yim, D. Tischer, T. Broderick, D. Baker, R. Barzilay, and T. Jaakkola, Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem, in *Proc. 11th Int. Conf. Learning Representations* (*ICLR*), Kigali, Rwanda, 2023.

[44] R. Yang, P. Srivastava, and S. Mandt, Diffusion probabilistic modeling for video generation, arXiv preprint arXiv: 2203.09481, 2022.

[45] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, Palette: image-to-image diffusion models, arXiv preprint arXiv: 2111.05826, 2021.

[46] F. Yang, Q. Zhai, X. Li, R. Huang, A. Luo, H. Cheng, and D. P. Fan, Uncertainty-guided transformer reasoning for camouflaged object detection, in *Proc. 2021 IEEE/CVF Int. Conf. Computer Vision* (*ICCV*), Montreal, Canada, 2022, pp. 4126–4135.

[47] M. Zhuge, X. Lu, Y. Guo, Z. Cai, and S. Chen, CubeNet: X-shape connection for camouflaged object detection, *Pattern Recognit.*, vol. 127, pp. 108644, 2022.

[48] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Y. Lo et al., Segment anything, arXiv preprint arXiv: 2304.02643, 2023.

[49] G. P. Ji, D. P. Fan, P. Xu, M. -M. Cheng, B. Zhou, and L. Van Gool, SAM struggles in concealed scenes—Empirical study on “segment anything”, arXiv preprint arXiv: 2304.06022, 2023.

[50] Q. Zhang, G. Yin, Y. Nie, and W. S. Zheng, Deep camouflage images, *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 12845–12852, 2020.

[51] Y. Li, W. Zhai, Y. Cao, and Z. J. Zha, Location-free camouflage generation network, *IEEE Trans. Multimedia*, pp. 1–14, 2022.

[52] T. N. Le, T. V. Nguyen, Z. Nie, M. T. Tran, and A. Sugimoto, Anabranch network for camouflaged object segmentation, *Comput. Vis. Image Underst.*, vol. 184, no. , pp. 45–56, 2019.

[53] P. Skurowski, H. Abdulameer, J. Błaszczyk, T. Depta, A. Kornacki, and P. Kozieł, Animal camouflage analysis: CHAMELEON database, https://www.polsl.pl/rau6/chameleon-database-animal-camouflage-analysis/, 2018.

[54] Y. Lv, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D. P. Fan, Simultaneously localize, segment and rank the camouflaged objects, in *Proc. 2021 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Nashville, TN, USA, 2021, pp. 11586–11596.

[55] W. Liu, X. Shen, C. M. Pun, and X. Cun, Explicit visual prompting for low-level structure segmentations, in *Proc. 2023 IEEE/CVF Conf. Computer Vision and Pattern Recognition* (*CVPR*), Vancouver, Canada, 2023, pp. 19434–19445.

[56] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, Improved techniques for training GANs, in *Proc. 30th Conf. Neural Information Processing Systems* (*NIPS 2016*), Barcelona, Spain.

[57] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, Learning to detect salient objects with image-level supervision, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*). Honolulu, HI, USA, 2017, pp. 3796–3805.

[58] J. Shi, Q. Yan, L. Xu, and J. Jia, Hierarchical image saliency detection on extended CSSD, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 717–729, 2016.

[59] C. Xia, J. Li, X. Chen, A. Zheng, and Y. Zhang, What is and what is not a salient object? Learning salient object detector by ensembling linear exemplar regressors, in *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, HI, USA, 2017, pp. 4399–4407.