Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052656

ARTICLE

Floating Waste Discovery by Request via Object-Centric Learning

Bingfei Fu*

School of Computer Science, Fudan University, Shanghai, 200438, China

*Corresponding Author: Bingfei Fu. Email: fubf2314852600@gmail.com

Received: 10 April 2024 Accepted: 11 June 2024 Published: 18 July 2024

ABSTRACT

Discovering floating wastes, especially bottles on water, is a crucial research problem in environmental hygiene. Nevertheless, real-world applications often face challenges such as interference from irrelevant objects and the high cost associated with data collection. Consequently, devising algorithms capable of accurately localizing specific objects within a scene in scenarios where annotated data is limited remains a formidable challenge. To solve this problem, this paper proposes an object discovery by request problem setting and a corresponding algorithmic framework. The proposed problem setting aims to identify specified objects in scenes, and the associated algorithmic framework comprises pseudo data generation and object discovery by request network. Pseudo-data generation generates images resembling natural scenes through various data augmentation rules, using a small number of object samples and scene images. The network structure of object discovery by request utilizes the pre-trained Vision Transformer (ViT) model as the backbone, employs object-centric methods to learn the latent representations of foreground objects, and applies patch-level reconstruction constraints to the model. During the validation phase, we use the generated pseudo datasets as training sets and evaluate the performance of our model on the original test sets. Experiments have proved that our method achieves state-of-the-art performance on Unmanned Aerial Vehicles-Bottle Detection (UAV-BD) dataset and self-constructed dataset Bottle, especially in multi-object scenarios.

KEYWORDS

Unsupervised object discovery; object-centric learning; pseudo data generation; real-world; object discovery by request

1 Introduction

Localizing floating waste in nearshore environments, such as rivers, lakes, and sandy areas, holds significant importance in environmental monitoring, with broad applications across various real-world scenarios [1–4]. Although UAV-BD [5] and Floating Waste (FloW) [6] datasets have gathered floating waste data from various scenes, they still fall short of encompassing all types of floating waste encountered in real-world scenarios. Given the significant variations in the shapes of floating waste and the diverse nearshore environments, these tasks often require the collection of field data, leading to inevitable data collection costs.

To reduce these high costs, some researchers have focused on unsupervised object detection or unsupervised object discovery methods, utilizing the robust generalization capabilities of unsupervised approaches to identify and locate floating waste. Considering that these tasks emphasize the localization of waste over its specific categorization, this paper adopts the research paradigm of unsupervised object discovery to address this issue.

Unsupervised object discovery aims to find potential objects by analyzing visual features. Recent unsupervised object discovery methods use pre-trained models to extract image features and utilize feature similarity for object localization [7,8]. Some researchers develop graph construction methods [9,10], while others utilize trainable self-supervised structures to refine object features [11,12]. It is noteworthy that the majority of existing methods tend to discover all objects in an image. However, practical applications often necessitate targeted object discovery; meanwhile, real-world scenes frequently encompass many irrelevant objects [13]. These present additional interference with the object discovery capacity of prevailing unsupervised methods, thereby challenging models when tasked with identifying specific objects.

To tackle the challenges above, this paper introduces the problem formulation of object discovery by request, designed to meet the practical requirements for identifying specific objects in the real world. Moreover, we provide a corresponding algorithmic framework. Object discovery by request aims to identify specific objects in actual scenes while disregarding other objects present in the scene. The proposed algorithmic framework consists of pseudo data generation and object discovery by request network. Pseudo data generation commences by augmenting a limited set of object samples, incorporating cropping, deformation, and color variation operations, thereby simulating various objects that may exist in actual environments. These augmented objects are subsequently blended with background images to enhance the fidelity of the synthesized images, making them closely resemble real-world scenes. For object discovery by request network, we propose three modules: Suspected Foreground Discovery (SFD), Object-Centric Learning (OCL) and Background Representation Learning (BRL). While SFD module extracts the image feature, OCL module learns the feature of the foreground object by request, and BRL module completes the background feature. Experimental results demonstrate that this approach can achieve state-of-the-art performance on UAV-BD and Bottle dataset, especially for multi-object discovery tasks (shown in Fig. 1). We summarized the contributions of this work as follows:

1) In response to the practical demand for discovering floating waste in nearshore environments, this paper proposes an object discovery by request problem setting, which enables the model to locate multiple specific objects in natural scenes under an unsupervised paradigm. Moreover, this paper proposes a corresponding algorithm framework, including pseudo data generation and object discovery by request network.
2) This paper introduces a simple and effective pseudo dataset generation method capable of generating a synthetic dataset suitable for model training. This method is achieved by augmenting a small quantity of foreground object data and employing a synthesis approach that combines foreground object data with background images.
3) This paper proposes an object discovery by request network consisting of SFD, OCL, and BRL modules. Based on these three designs, our model can learn specified object representations in a latent space while preventing the model from confusing background and irrelevant objects.

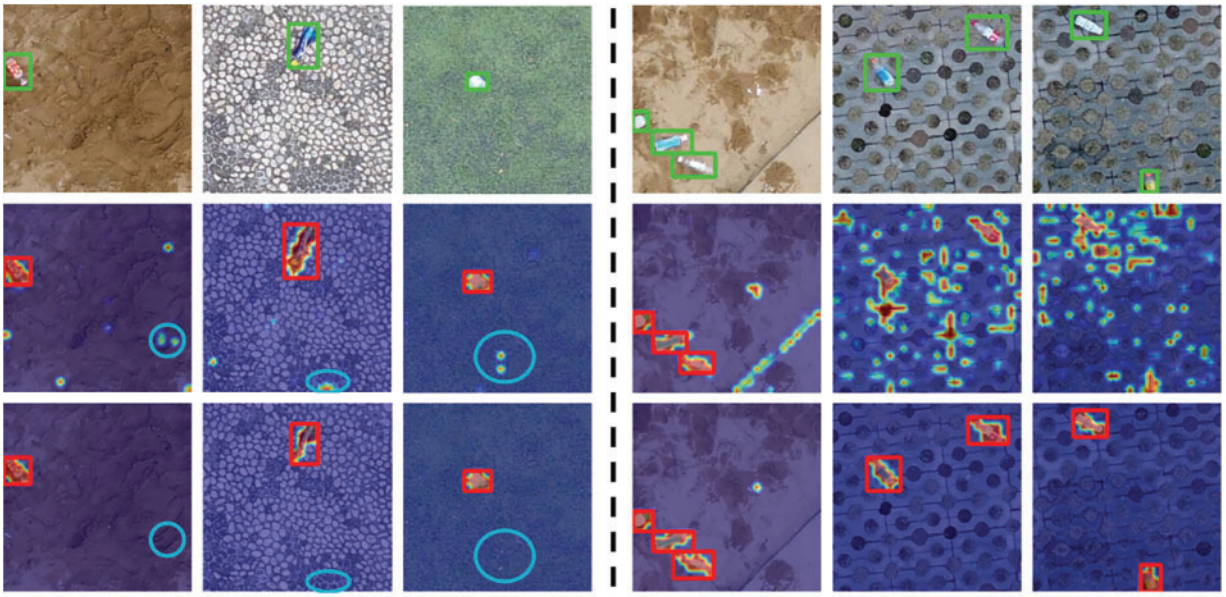

Figure 1: Comparison of the state-of-art method called FOUND (second line) and ours (third line) on UAV-BD with single (left) and multi (right) objects. We circled wrong activations of FOUND in blue

2 Related Works

2.1 Unsupervised Object Discovery & Detection

Unsupervised object discovery and detection methods aim to localize objects within images in the absence of explicit supervision [14,15]. These researches typically involve preprocessing steps, using feature extractor to obtain visual feature from images. Early methods [16] relied on machine learning algorithms such as Support Vector Machines (SVM) [17], k-Nearest Neighbors (kNN) [18], and decision trees [19] to identify objects based on pixel-level correlations within images. With the advent of Convolutional Neural Networks (CNNs), researchers [20–22] began utilizing neural network to extract hierarchical features from images. Some researches employed CNNs to generate candidate bounding boxes [23–25], which were evaluated for their likelihood of containing foreground objects. Recently, the introduction of ViT (Vision Transformer) architecture has led to novel ideas in unsupervised object discovery and detection tasks [26–28]. ViT models leverage self-attention mechanism to capture long-range dependencies in tokens, allowing for feature extraction across different scales [29]. Current researches, such as LOST (Localizing Objects with Self-Supervised Transformers) [9] and Cut-and-LEaRn (CutLER) [30], leverage the property of self-supervised method to localize objects in different datasets.

However, existing researches often focus on identifying all objects present in images, neglecting the specific requirements of real-world applications where only certain objects are of interest [31,32]. To solve this problem, LOST [9] pioneered the introduction of graph construction, leveraging normalized cut (N-cut) to locate interested objects within images. TokenCut [10] improved N-cut mechanism, enhancing the attention map's focus on objects that significantly differ from the scene. FOUND [12] proposed a learnable network architecture, further increasing the distinction between background and interested objects. Nevertheless, these methods still have difficulty in achieving the goal of customizing objects of interest. To address this gap, recent research [31] has introduced the concept of

"request", which aims to localize specific objects in images based on user-defined criteria and leverage multiple iterations to achieve targeted object recognition. By associating this concept with our practical application scenarios, this paper proposes a problem formulation of object discovery by request and a corresponding algorithmic framework.

2.2 Object-Centric Learning

To localize specified objects within images, we explore a feature learning method capable of learning representations for individual objects. Object-centric learning (OCL) method is acknowledged as a form of unsupervised learning that specializes in acquiring representations of individual objects within images [33,34]. One prevalent architecture within OCL method is slot attention [33], which projects input images into a lower-dimensional latent space and learns object representations through image reconstruction. Early research [34] has demonstrated the capacity to acquire representations of individual objects on toy datasets, such as Compositional Language and Elementary Visual Reasoning (CLEVER) and Multipurpose Motion and Video (MOVi) datasets. Recent developments in OCL research have focused on acquiring object representations in real-world settings. For instance, DINOSAUR [35], built upon the pre-trained DINO [36] model, enables models to learn object-centric latent space representations under an unsupervised paradigm and accomplish object segmentation tasks. Bi-level Optimized Query Slot Attention (BO-QSA) [37] enhances the slot attention structure to facilitate object segmentation in single-object scenarios. Therefore, this paper incorporates the OCL method into the object discovery by request framework, enabling the model to autonomously learn representations of objects of interest within images under an unsupervised paradigm.

3 Problem Formulation of Object Discovery by Request and Pseudo Dataset Generation

3.1 Object Discovery by Request Problem Setting

As one of the foundational research areas in computer vision, object discovery methods find extensive applications in real-world scenarios. Presently, most approaches globally discover all objects within a scene, potentially leading to oversights in multi-object environments. Furthermore, practical applications often demand finer-grained and dynamically evolving object categories, posing challenges for existing methods to maintain robust performance. In light of the constraints inherent in practical object discovery tasks, we propose two assumptions:

- Limited Scene Variability: Within a given practical application context, despite potential dynamic changes in the scene's objects, the overarching scene categories and stylistic attributes remain relatively consistent.
- Limited Object Category Variability: In practical application contexts, the requisite object categories may exhibit variability or expansion, yet the overall object category maintains a degree of stability.

Based on the aforementioned assumptions and in response to the variability of object categories in real-world scenarios, this paper proposes the problem formulation of object discovery by request to address the challenges of object discovery in authentic scenes. Object discovery by request aims to discover the specified objects in real-world under the limitation of scene collection. Simultaneously, this paper introduces the corresponding framework for object discovery by request, comprising two key components: pseudo data generation and object discovery by request network. Pseudo data generation uses a limited number of object samples and scene images to synthesize dataset, while the algorithm employs an object-centric learning structure within an unsupervised paradigm to learn the

representations for specified object. In cases where the appearance and categories of objects change, this method can swiftly generate datasets corresponding to the variable categories using data synthesis methods and retrain the model within an unsupervised paradigm.

Detailed explanations of pseudo data generation and object discovery by request network will be provided in Sections 3.2 and 4, respectively.

3.2 Pseudo Dataset Generation

Fig. 2 shows the complete process of pseudo data generation, including three different augment rules for background images and foreground objects. The synthetic images in pseudo dataset are represented as $I = \boldsymbol{f}(bck) + \beta \times \boldsymbol{f}(obj)$, where *bck* and *obj* represent the samples of background images in real-world and the samples of foreground objects need to be discovered, $\boldsymbol{f}$ denotes different data augmentation rules and β represents the the opacity of foreground objects. Our pseudo data generation method first applies data augmentation separately to *obj* and *bck*, then combine them together using the hyperparameter $\beta = 0.6$. The bottom of Fig. 2 displays three different data augmentation rules applied to foreground objects and background images, respectively.

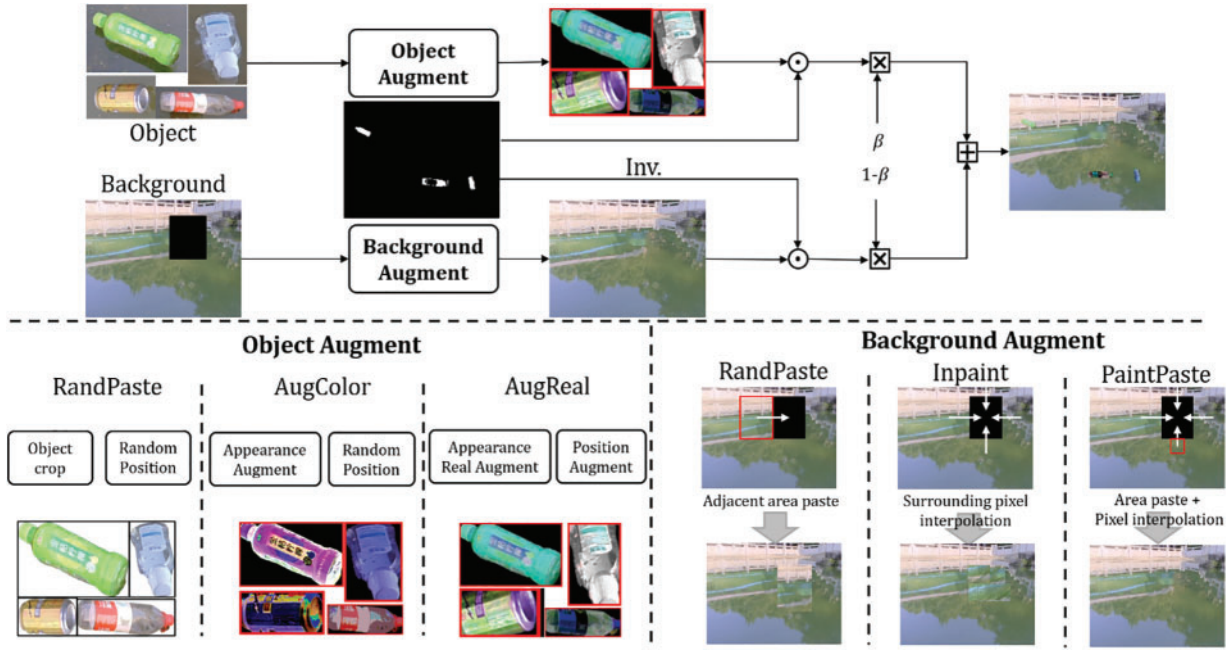

Figure 2: Illustration of pseudo data generation. The upper part shows the complete process of generation and the lower part presents different augment rules of foreground objects (RandPaste, AugColor and AugReal) and background images (RandPaste, Inpaint and PaintPaste)

For foreground objects, three augmentation techniques are employed: RandPaste, AugColor, and AugReal (proposed in this paper). RandPaste involves two operations, object cropping and random positioning, which crop object samples and paste them randomly onto scene images for data synthesis. AugColor includes both appearance augmentations [38,39] and random positioning to enhance the visual diversity of object samples, improving the extensibility of the synthesized data. AugReal builds upon AugColor by further augmenting the appearance and deformity of object samples to more closely

mimic the natural appearance of objects in real-world scenes. Additionally, it expands the positioning of object samples to better align with real-world settings.

For background images, given that scene samples may contain distracting objects, this method applies a cropping operation to remove these objects and employs three rules for image completion: RandPaste, Inpaint [40], and PaintPaste (proposed in this paper). RandPaste resizes the surrounding regions and fills the cropped area with them. Inpaint uses interpolation methods based on surrounding pixels to complete the cropped area. PaintPaste combines the first two approaches by initially filling the cropped area with random surrounding regions and then using interpolation methods to smooth out the filled area.

Utilizing the aforementioned pseudo data generation method, we can quickly generate the required pseudo datasets across various real-world scenarios, alleviating the costs associated with data collection. To validate the performance of our algorithm framework, this paper constructs corresponding pseudo datasets and testing environments based on the UAV-BD [5] dataset and a self-collected Bottle dataset, as shown in Table 1. For the UAV-BD dataset, we randomly selected 200 scene samples and 10 object samples from 16,258 images in the training set for pseudo data synthesis, resulting in a total of 5000 synthesized images. The test environment was evaluated using the provided UAV-BD test set, which comprises 6944 bottles across 5081 images. For the self-collected Bottle dataset, we randomly sampled 58 scene samples in different environments and 5 bottle samples to generate 790 pseudo data images, while the remaining 479 images were used as the test set.

Table 1: The parameters of scene/object samples, amount of pseudo datasets and test sets on UAV-BD and bottle dataset

Dataset name	Amount of scene sample	Amount of object sample	Amount of synthesized images	Amount of test set
UAV-BD	200	10	5000	5081
Bottle	58	5	790	479

4 Object Discovery by Request Network

The overall architecture of our method is shown in Fig. 3. The framework contains a Vision Transformer (ViT) based feature extractor, Suspected Foreground Discovery (SFD) module, Object-Centric Learning (OCL) module and Background Representation Learning (BRL) module, can be used to discover specified objects in real-world under unsupervised learning.

This section is organized as follows: we first briefly review the ViT and SFD module in Section 4.1. Then we describe the OCL and BRL module in Sections 4.2 and 4.3. Finally, we introduce the training & evaluation pipelines in Section 4.4.

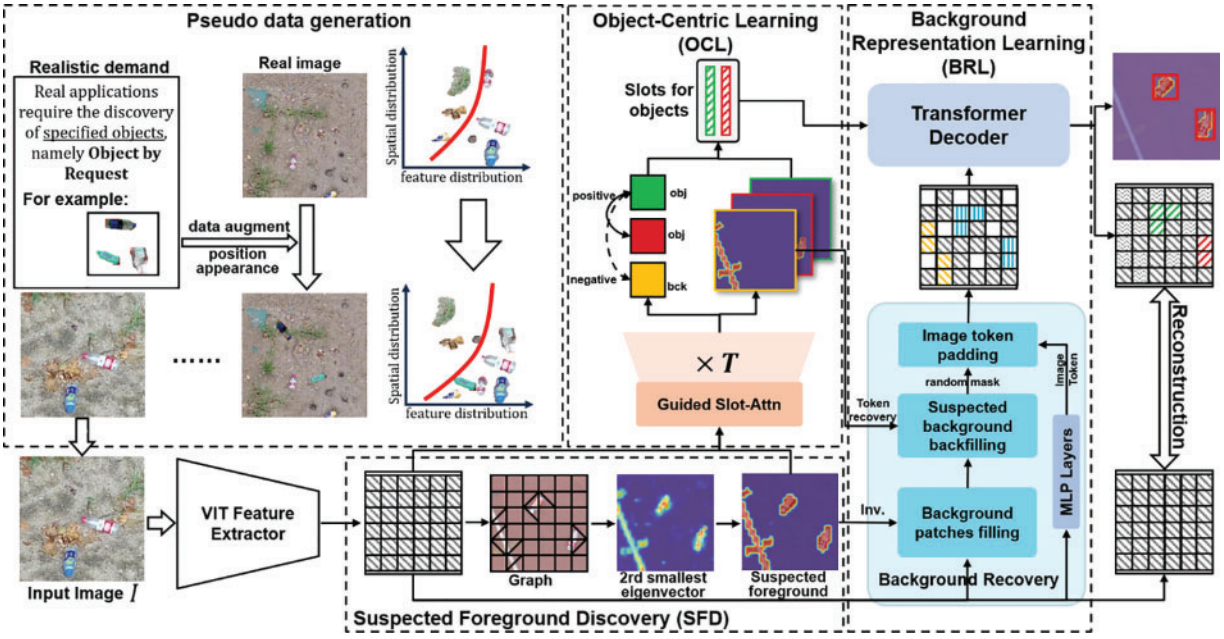

Figure 3: Illustration of the whole architecture of our method. Using the synthetic images from Pseudo data generation, SFD module initially divides the foreground areas, OCL module learns the representation of objects by request, and BRL module reconstructs the whole image to achieve unsupervised object discovery

4.1 ViT and SFD Module

The ViT module, originally introduced by [36], employs a transformer architecture to process images by segmenting them into non-overlapping patches. In our method, we utilize a pre-trained ViT model trained on ImageNet as a feature extractor. This model partitions the input image I of dimensions $H \times W$ into $K \times K$ non-overlapping image patches. Leveraging the transformer network, we derive $N = HW/K^2$ tokens (O_n) representing patches and a class token (O_{cls}) containing global representation. Our ViT-based feature extractor comprises 12 stacked encoder blocks, each comprising a feedforward network and a multi-head self-attention mechanism. The tokens extracted from the final block serve as inputs for the SFD, OCL, and BRL modules.

The SFD module draws inspiration from current researches (Ncut [41] and TokenCut [10]), utilizing N tokens extracted by ViT to construct a graph $G = (V \times E)$, where N tokens serve as nodes and E represents the similarity matrix among N tokens. In order to partition V into two subsets, background B and foreground O, Ncut [41] proposes the following energy function E:

$$E = \frac{\boldsymbol{S}(O,B)}{\boldsymbol{S}(O,V)} + \frac{\boldsymbol{S}(O,B)}{\boldsymbol{S}(B,V)} \tag{1}$$

where the function $\boldsymbol{S}$ measures the similarity between two sets. Through the minimization of the objective function E, the score of each token is derived based on the second smallest eigenvector. To ensure the comprehensive allocation of all objects to the foreground set, we implement a secondary top-k selection procedure based on token scores, thereby ensuring the complete classification of all objects into foreground region $Mask_{fore}$.

4.2 Object-Centric Learning Module

For OCL module (shown in Fig. 4), our method employs the tokens extracted by the ViT module as input, enabling the model to learn the latent space representation of foreground objects within the scene. Unlike previous object-centric methods that directly utilize all tokens as input, our OCL module, considering the complexity of visual features in real-world scenes, initially filters all tokens through the SFD module. This selection process retains tokens that potentially contain foreground objects as input, facilitating the model's expedited learning of the latent space representation of objects under guidance.

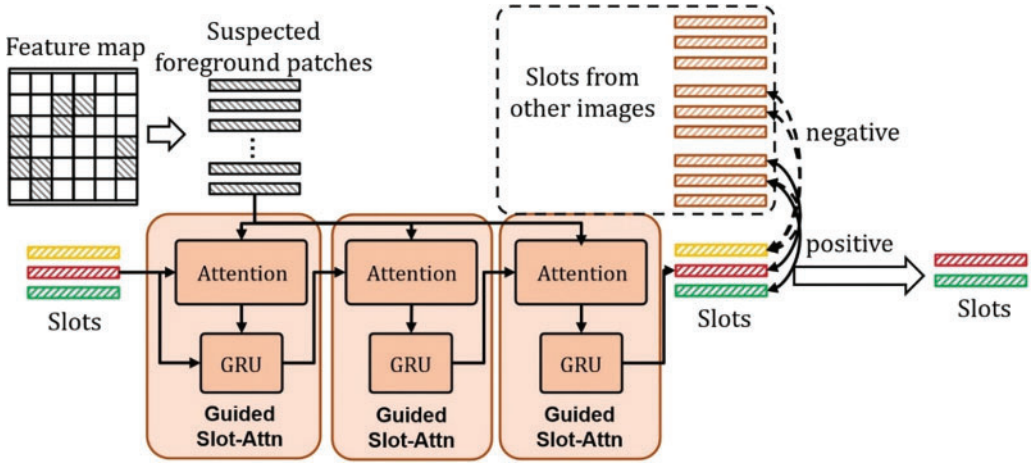

Figure 4: Illustration of OCL design, including guided slot-attention and contrastive learning mechanism for slots

We input patches potentially containing foreground objects into a guided slot-attention block upon obtaining them. This block encompasses Attention and Gated Recurrent Unit (GRU). The Attention component computes the dot product between patches to derive a relevance matrix. The GRU module, proposed by [33], employs a learnable recursive function to update slots, thereby individually disentangling objects in the image into separate slots. The specific formulation is as follows:

$$S^t = z \times S^t + (1 - z) \times \textbf{Update}\left(\textbf{Attn}\left(O_{fore}\right)\right) \tag{2}$$

where S represents the vector representation of slots, t denotes the number of iterations for guided Slot-Attn, z represents the adaptable gating parameters, O_{fore} means tokens representing foreground objects, **Attn** represents the self-attention mechanism and **Update** denotes the function that aligns the correlation matrix of patches with the dimensions of slots. Through multiple iterations, the Guided Slot-Attn module maps the suspected foreground patches obtained by SFD module to different regions in the latent space and obtains corresponding latent representations S.

Since the input patches contain some background and irrelevant objects, Guided Slot-Attn can only decouple features into different slots but cannot identify whether the slot corresponds to the object to be discovered by request. Therefore, we introduce a self-supervised mechanism for the slots by computing the Info Noise Contrastive Estimation (InfoNCE) loss function between the current

slot and slots from other images, as detailed below:

$$loss_{infonce} = -\sum_{b=0}^{B}\sum_{i=0}^{L} log\frac{exp\left(S_{bi} \times S_{bi}/\tau\right)}{\sum_{j=0}^{L} exp\left(S_{bi} \times S_{bj}/\tau\right)} \tag{3}$$

where τ denotes the temperature coefficient, B represents the batch size for training model, and L signifies the maximum number of potentially present objects in images. S_{bi} corresponds to the slots of the current image, while S_{bj} denotes the set of slots for all images in the same batch. By minimizing loss inf $loss_{nce}$, this model further amplifies the distributional differences in the latent space between the specific objects to be discovered and other objects/background.

To find out the slots representing background and irrelevant objects, we use the cosine similarity to measure the distances between slots:

$$Dis_i = \frac{1}{BL-1}\sum_{j=0,\ i\neq j}^{BL} \left\| S_i,\ S_j \right\|_2^2 \tag{4}$$

where Dis_i means the average distances between S_i and other slots in one batch. We use the average value $\overline{Dis_i} = 1/BL\sum_{j=0}^{BL} Dis_j$ to measure whether slots represent specific objects that need to be discovered, and filter out slots that represent background and irrelevant objects in the subsequent image reconstruction process, ensuring that the model focuses exclusively on the objects by request and better learns their latent space representation.

4.3 Background Representation Learning Module

For BRL module (shown in Fig. 5), tokens from various modules are sequentially embedded into image patches, which are then utilized for image reconstruction via Transformer Decoder. Using cross-attention mechanism, we build a mapping between slots and patches, enabling our model to utilize slots representing objects for patch reconstruction. Through reconstruction loss, our model discovers and reconstructs the targeted objects.

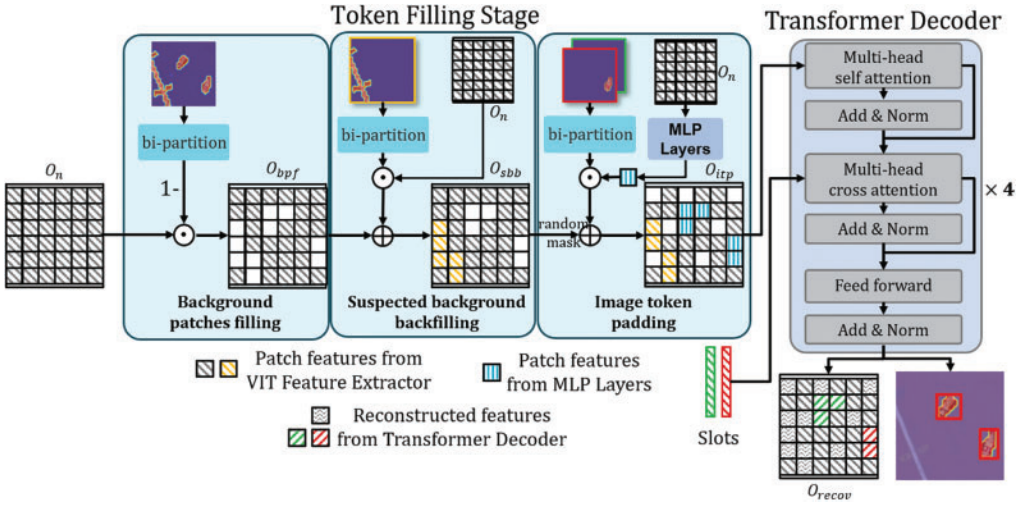

Figure 5: Illustration of BRL module, including background patch filling, suspected background backfilling, image token padding and transformer decoder

During token filling stage, we proceed with the following three steps: background patches filling, suspected background backfilling, and image token padding. In the first step, tokens identified as background in SFD module are filled into image patches. Secondly, by computing the *Dis* for each slot, areas corresponding to slots representing background/irrelevant objects are backfilled using patch features extracted by ViT. Thirdly, the remaining regions are filled with image tokens representing global features extracted by Multi-layer Perceptron (MLP) layers. The mathematical formulations of the three steps are delineated as follows:

$$
\begin{aligned}
O_{bpf} &= O_n \times \left(1 - \boldsymbol{f}_{\boldsymbol{bi}}\left(Mask_{fore} \times O_n\right)\right) \\
O_{sbb} &= O_{bpf} + O_n \times \boldsymbol{f}_{\boldsymbol{bi}}\left(\mathbf{Attn}\left(slot_{-}, O_{fore}\right)\right) \\
O_{itp} &= \boldsymbol{f}_{\boldsymbol{rand}}\left(O_{sbb}\right) + \mathbf{MLP}\left(O_n\right) \times \left(\boldsymbol{f}_{\boldsymbol{bi}}\left(O_{fore}\right) - \boldsymbol{f}_{\boldsymbol{bi}}\left(Attn\left(slot_{-}, O_{fore}\right)\right)\right)
\end{aligned}
\tag{5}
$$

where $\boldsymbol{f}_{\boldsymbol{bi}}$ performs the binary matrix operation on input features and $\boldsymbol{f}_{\boldsymbol{rand}}$ represents the drop out operation on patch features. **Attn** computes the correlation matrix between slots and O_{fore}. slot represents the sequence of slots where $Dis > \overline{Dis}$, while **MLP** refers to a multi-layer convolutional network used for extracting global image representations. Through these operations, our method reduces the difficulty of model reconstruction and fills foreground object regions with Image tokens representing global features. This enables our model to utilize slots to guide the reconstruction of foreground objects in Transformer Decoder, facilitating better learning of the features of objects by request.

For Transformer Decoder, we take O_{itp} and S as inputs, utilizing S to guide this module to reconstruct representations O_{recov} that conform to foreground objects, and calculates L2 loss with patch features O_n from the pre-trained model, as described below:

$$
\begin{aligned}
loss_{recov} &= \left\| O_{recov},\ O_n \right\|_2^2 \\
loss_{final} &= loss_{recov} + \alpha loss_{infonce}
\end{aligned}
\tag{6}
$$

where $loss_{final}$ is the final loss for model training, which includes the self-supervised contrastive loss $loss_{nce}$ from SFD module and the patch reconstruction loss $loss_{recov}$ from BRL module, with a hyperparameter $\alpha = 0.2$. With the aforementioned objective loss function, our method can further distinguish slots while reconstructing features, guiding the model to learn the representations of foreground objects by request more efficiently.

4.4 Training and Inference Pipeline

We freeze the pre-trained ViT module parameters as in previous works [9,10,12]. During the training stage, all the tokens from ViT module are first fed into SFD module to obtain suspect foreground area $Mask_{fore}$. Afterward, tokens representing foreground objects are fed into OCL module, where objects are learned into different slots through slot attention [33], and a self-supervised contrastive mechanism is used to determine whether slots belong to the object of interest. On the other hand, tokens representing background are input into BRL module, undergo token filling, and along with the slots, are input into Transformer Decoder. Reconstructed patch features are ultimately obtained, and reconstruction loss is computed based on tokens obtained from pre-trained models.

During the inference stage, we use the slot-attention maps as foreground offsets to refine the heatmaps of foreground objects. The refined heatmaps are input into the multi-object localization head to derive the ultimate bounding boxes. It is noteworthy that, given the unsupervised nature of the

object discovery task, which aims at localizing foreground objects irrespective of their categories, the resulting bounding boxes are inherently class-agnostic.

5 Experiment

In the experiment section, this paper outlines the evaluation metrics and implementation details in Section 5.1. Then, we validate the challenges inherent in the object discovery by request problem and the effectiveness of pseudo data generation in Section 5.2. Next, this paper presents experimental comparisons and visualization results for the object discovery by request problem setting, along with comparative methods in Section 5.3. Furthermore, we verify the effectiveness of different data augmentations and each module in Section 5.4.

5.1 Evaluation Metrics and Implementation Details

Evaluation Metrics. We conducted evaluations on two benchmark datasets employing two common metrics: mean Intersection over Union (mIoU) and Correct Localization (CorLoc), as delineated below:

$$\text{mIoU} = \frac{\text{TP}}{\text{FP} + \text{FN} + \text{TP}}$$

$$\text{Corloc} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{7}$$

where the objects by request are considered positive samples while all other objects and background are regarded as negative samples. TP represents the count of correctly identified positive samples, FP denotes the number of false positives, TN signifies the count of correctly identified negative samples, and FN indicates the number of missed positive samples. For the CorLoc metric, we refine it into two measurements: $\text{CorLoc}_{\text{all}}$ and $\text{CorLoc}_{\text{multi}}$. $\text{CorLoc}_{\text{all}}$ represents the localization measurement across the entire test dataset, while $\text{CorLoc}_{\text{multi}}$ represents the localization measurement in multi-object environment.

Given the objective of addressing real-world object discovery by request in this paper, our proposed method and the comparative approaches are trained on synthetic datasets and evaluated for performance on the test sets in real scenes.

Implementation Details. Our method is implemented on PyTorch framework and trained on one NVIDIA TITAN X GPU, with the system environment running on Ubuntu 18.04. The training parameters of our method are shown in Table 2. For training, we use the pre-trained DINO [36] model as ViT backbone, utilizing an AdamW optimizer with 0.05 weight decay and 2e-4 learning rate. For inputs, we resize them to 224×224 resolution and set the batch size to 8. Regarding model parameters, we configured the top-k filtering ratio in SFD module to 25%, the maximum number of slots in OCL module to 4, and the iteration number for guide slot-attention to 3. Additionally, we set the random mask in BRL module to 20% and the number of transformer blocks to 4.

Table 2: The parameters of training process of our method

Parameters	Epoch	Learning rate	Image size	Batch size	Tok-k ratio in SFD	Slot number	Random mask in BRL
Training setting	100	2e-4	224	8	25%	4	20%

5.2 Challenges in Object Discovery by Request & Effectiveness on Pseudo Dataset Generation

In this section, we quantitatively demonstrate the challenges of object discovery by request and underscore the imperative nature of pseudo dataset generation to tackle this problem.

In Table 3, experiment results of YOLO [42] and Bottle [12] models on UAV-BD dataset are presented under both general and object discovery by request settings. Under the general setting, models are trained on the train set and evaluated on the test set. However, under the object discovery by request setting, models are trained on a pseudo dataset composed of limited data samples and scene images synthesized from real-world scenarios and then evaluated on the test set in the real world. As a supervised model, YOLO experiences an 11.8% decrease in detection performance for specific objects, while as an unsupervised paradigm, FOUND witnesses a 26.9% decrease in the discovery performance of specific objects. This indicates the challenging nature of the proposed object discovery by request problem.

Table 3: Experiments on YOLO and FOUND models under general settings (odd rows) and object discovery by request settings (even rows)

Model	Training setting	$\text{CorLoc}^{50}_{\text{all}}$	$\text{CorLoc}^{75}_{\text{all}}$
YOLO	UAV-BD train set	92.9	87.4
YOLO	Pseudo dataset	85.3 (7.6↓)	34.2 (53.2↓)
FOUND	UAV-BD train set	83.5	56.0
FOUND	Pseudo dataset	56.6 (26.9↓)	21.9 (34.1↓)

Table 4 presents the results of YOLO and FOUND models under object discovery by request setting, which only pre-trained on ImageNet or fine-tuned with pseudo dataset. The experiments on UAV-BD dataset shows that both supervised and unsupervised-trained models exhibit significant improvement in the discovery performance of specific objects after fine-tuning with pseudo dataset (YOLO: 41.6%↑, FOUND: 33.5%↑), which reveals the effectiveness of pseudo dataset generation in object discovery by request research.

Table 4: Experiments on YOLO and FOUND models with fine-tune operation (even rows) or not (odd rows) under object discovery by request settings

Model	Fine-tune setting	$CorLoc^{50}_{all}$	$CorLoc^{75}_{all}$
YOLO	Without fine-tune	43.7	5.8
YOLO	Fine-tune	85.3 (41.6↑)	34.2 (28.4↑)
FOUND	Without fine-tune	23.1	1.5
FOUND	Fine-tune	56.6 (33.5↑)	21.9 (20.4↑)

5.3 Results and Comparison on Object Discovery by Request

In this section, we compare our method with several state-of-the-art unsupervised approaches (LOST [9], TokenCut [10] and FOUND [12]). Table 5 presents the comparative results across UAV-BD and Bottle datasets. On UAV-BD dataset, our method exhibits improvements of 2.1% and 2.9% in the metrics of $CorLoc^{50}_{all}$ and $mIoU^{50}$. Notably, there is a significant enhancement of 3.7% observed in $CorLoc^{50}_{multi}$ metric, which focuses on multi-object localization. On Bottle dataset, our method demonstrates significant improvements compared to state-of-the-art methods ($CorLoc^{50}_{all}$: 2.1%↑, $mIoU^{50}$: 3.5%↑). These results underscore the effectiveness of our method in discovering specified objects within real-world scenarios under unsupervised setting. Even in scenarios with multiple target objects present, our method maintains high performance on object discovery.

Table 5: Experiments on YOLO and FOUND models under general settings (odd rows) and object discovery by request settings (even rows)

Method	Training setting		UAV-BD			Bottle		
Model	Network weight	Fine-tune	$CorLoc^{50}_{all}$	$CorLoc^{50}_{multi}$	$mIoU^{50}$	$CorLoc^{50}_{all}$	$CorLoc^{50}_{multi}$	$mIoU^{50}$
ViT	Train on ImageNet	w/o fine-tune	38.2	37.7	34.5	21.9	19.8	19.2
LOST	ViT model	Unable	42.5	41.7	40.1	23.5	21.4	19.4
Token Cut	ViT model	Unable	49.3	45.8	42.9	35.6	32.6	30.5
FOUND	DINO model	Pseudo dataset	56.6	51.6	50.2	37.3	34.6	32.4
Ours	DINO model	Pseudo dataset	**58.6**	**56.2**	**53.6**	**38.1**	**36.5**	**34.0**

Fig. 6 illustrates the visualization results of our method on UAV-VD and Bottle datasets. Our method adeptly localizes the objects by request (bottles and cans) across various real-world environments, including foliage, sandy terrain, and bodies of water. Furthermore, in complex multi-object scenarios, our method reliably accomplishes the localization of multiple target objects.

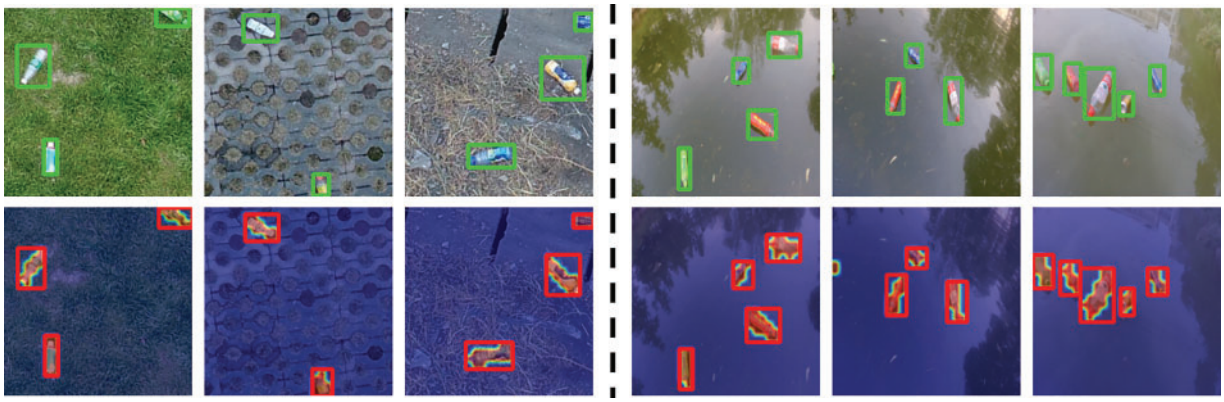

Figure 6: Visualization results on UAV-BD and bottle dataset

5.4 Ablation Studies

In this section, we evaluate the effectiveness of pseudo dataset, the inner modules of our method and hyper-parameters in network on UAV-BD dataset.

Effectiveness on Pseudo Dataset. To independently validate the effectiveness of background augmentation and object augmentation, we fixed the augmentation rules of object and background separately and generated corresponding pseudo datasets. Subsequently, YOLO models are trained on these pseudo datasets and evaluated on the test set of UAV-BD. Table 6 presents the results of YOLO models trained on the pseudo datasets with background rule RandPaste, while Table 7 shows the results of YOLO models trained on the pseudo datasets with object rule AugReal. Compared to the random pasting operation (RandPaste) on object and background, our proposed augmentation rules (AugReal and PaintPaste) significantly improve the performance of YOLO model on the test set of UAV-BD, with $\text{CorLoc}^{50}_{\text{all}}$ increasing from 51.8% to 85.3%.

Table 6: Experiments of different foreground augment rules (background rules fixed RandPaste)

Object augment	Background augment	$\text{CorLoc}^{50}_{\text{all}}$	$\text{CorLoc}^{75}_{\text{all}}$
RandPaste	RandPaste	51.8	7.1
AugColor	RandPaste	80.8	24.9
AugReal	RandPaste	81.3 (29.5↑)	28.3 (21.2↑)

Table 7: Experiments of different background augment rules (object rules fixed AugReal)

Object augment	Background augment	$\text{CorLoc}^{50}_{\text{all}}$	$\text{CorLoc}^{75}_{\text{all}}$
AugReal	RandPaste	81.3	28.3
AugReal	InPaint	84.6	31.3
AugReal	PaintPaste	85.3 (4.0↑)	34.2 (5.9↑)

Effectiveness on Modules. Next, we evaluate the inner modules of our proposed method. Tables 8–10 present effectiveness validation experiments for SFD, OCL, and BRL modules based on UAV-BD dataset. As shown in Table 8, compared to directly inputting patch features extracted by ViT model into subsequent modules, the proposed supervised foreground selection, combined with the graph construction of patch features and calculation based on the second smallest eigenvalue, can significantly improve the localization performance for specific objects (29.2%↑). Table 9 shows that comparing to the general object-centric method (slot-attention), the proposed guided slot-attention and its corresponding contrastive loss function significantly improve performance (13.2↑). Table 10 conducts the ablation experiment on BRL module. The results show that the proposed supervised background filling and image token padding have a certain improvement in performance (8.9%↑).

Table 8: Experimental verification on SFD module

Second smallest eigenvector	Suspected foreground	$CorLoc^{50}_{all}$	$CorLoc^{50}_{multi}$
−	−	29.4	21.9
+	−	43.1	40.6
+	+	58.6 (29.2↑)	56.2 (34.3↑)

Table 9: Experimental verification on OCL module

Guided slot-attention	Contrastive loss function	$CorLoc^{50}_{all}$	$CorLoc^{50}_{multi}$
−	−	45.4	41.9
+	−	55.6	55.1
+	+	58.6 (13.2↑)	56.2 (14.3↑)

Table 10: Experimental verification on BRL module

Suspected background backfilling	Image token padding	$CorLoc^{50}_{all}$	$CorLoc^{50}_{multi}$
−	−	49.7	46.5
+	−	57.4	55.9
+	+	58.6 (8.9↑)	56.2 (9.7↑)

Effect of Hyper-Parameters. Furthermore, we investigate the design of two hyperparameters: the slot number for SFD module and the mask rate of BRL module (shown in Fig. 7). Results show that when slot number = 4, mask rate = 25%, the performance of our method reaches its peak. Our analysis suggests that the slot number represents the maximum number of objects to be discovered in the image. When the slot number is less than the average number of objects to be discovered in the image, the model's performance deteriorates significantly. The mask rate indicates the proportion of background that the model needs to generate during the reconstruction process. Results indicate that appropriate background masking facilitates the model in better learning background features during the background completion process.

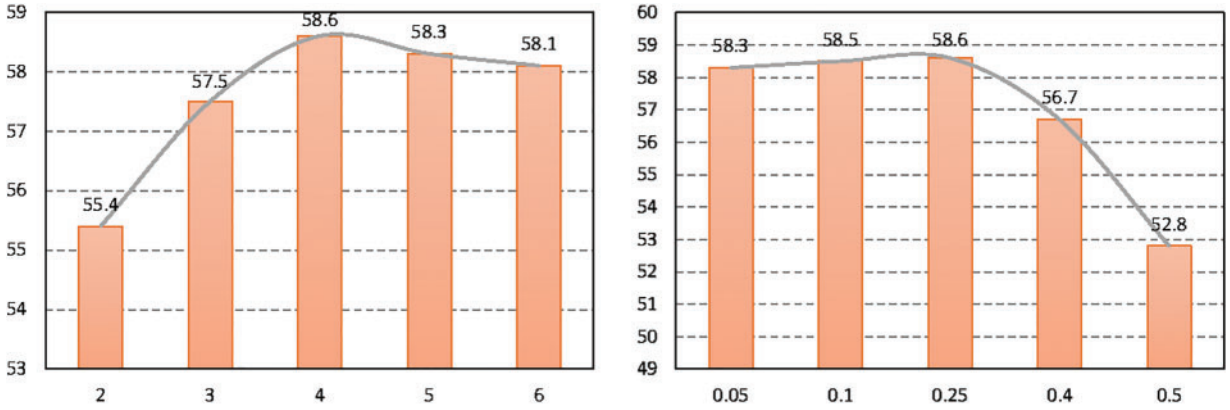

Figure 7: Ablation studies in slot number in OCL (left) and mask rate of BRL (right)

Regarding the real-time performance of our method, due to the use of N-cut (computed on Central Processing Unit, CPU) to assist in calculating the similarity between tokens, the actual computational efficiency of our model is lower than LOST [9] (close to Tokencut [10]). Compared to the methods like FOUND [12] that require training, our method has lower time costs during the training phase and lower real-time performance during the testing phase. However, this is not the main concern of our algorithm at present. By deploying N-cut to Graphics Processing Unit (GPU), we hope to significantly enhance the computational efficiency of our method.

6 Conclusion

This paper addresses the practical need to localize floating objects in real-world scenarios by delineating the problem formulation of object discovery by request and presenting a corresponding algorithmic framework. The object discovery by request problem aims to identify and localize specific objects within real-world scenes without supervision. The algorithmic framework encompasses two pivotal components: pseudo data generation and an object discovery by request network architecture. Pseudo data generation involves the creation of diverse synthetic images resembling real-world scenes, achieved through a limited set of object samples and scene images, along with data augmentation techniques for model training. The object discovery by request network architecture comprises three integral modules: SFD, OCL, and BRL. SFD module is responsible for extracting pertinent image features, OCL module focuses on learning the latent representation of foreground objects and discerning whether they are the specific objects of interest, while BRL module is tasked with reconstructing patch-level features and imposing constraints on model training. Experiments demonstrate that the proposed object discovery by request network, in conjunction with pseudo data generation, achieves state-of-the-art performance on both the UAV-BD dataset and a self-constructed Bottle dataset.

Acknowledgement: The authors would like to acknowledge the School of Computer Science, Fudan University for funding this work.

Funding Statement: The author received no specific funding for this study.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The author declares that the have no conflicts of interest to report regarding the present study.

References

[1] W. C. Li, H. F. Tse, and L. Fok, "Plastic waste in the marine environment: A review of sources, occurrence and effects," *Sci. Total Environ.*, vol. 566–567, pp. 333–349, Oct. 2016. doi: 10.1016/j.scitotenv.2016.05.084.

[2] A. Akib *et al.*, "Unmanned floating waste collecting robot," in *Proc. TENCON, 2019–2019 IEEE Region 10 Conf. (TENCON)*, Kochi, India, 2019, pp. 2645–2650.

[3] N. Ruangpayoongsak, J. Sumroengrit, and M. Leanglum, "A floating waste scooper robot on water surface," in *Proc. 17th Int. Conf. Control, Autom. Syst.*, Jeju, Republic of Korea, 2017, pp. 1543–1548.

[4] J. Niu, S. Gu, J. Du, and Y. Hao, "Underwater waste recognition and localization based on improved YOLOv5," *Comput. Mater. Contin.*, vol. 76, no. 2, pp. 2015–2031, Aug. 2023. doi: 10.32604/cmc.2023.040489.

[5] J. Wang, W. Guo, T. Pan, H. Yu, L. Duan and W. Yang, "Bottle detection in the wild using low-altitude unmanned aerial vehicles," in *Proc. 21st Int. Conf. Info. Fusion*, Cambridge, UK, 2018, pp. 439–444.

[6] Y. Cheng *et al.*, "Flow: A dataset and benchmark for floating waste detection in inland waters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 10953–10962.

[7] K. J. Hsu, Y. Y. Lin, and Y. Y. Chuang, "Co-attention cnns for unsupervised object co-segmentation," in *Proc. Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, 2018, pp. 748–756.

[8] Y. Z. Xu, C. Y. Chen, and C. T. Li, "SUVR: A search-based approach to unsupervised visual representation learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Rhodes Island, Greece, 2023, pp. 1–5.

[9] O. Simeoni *et al.*, "Localizing objects with self-supervised transformers and no labels," in *Proc. British Mach. Vis. Conf.*, Virtual, UK, 2021, pp. 1–16.

[10] Y. Wang, X. Shen, S. X. Hu, Y. Yuan, J. L. Crowley and D. Vaufreydaz, "Self-supervised transformers for unsupervised object discovery using normalized cut," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, New Orleans, LA, USA, 2022, pp. 14543–14553.

[11] Z. Li, L. Zhao, W. Chen, S. Yang, D. Xie and S. Pu, "Target-aware auto-augmentation for unsupervised domain adaptive object detection," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, Singapore, 2022, pp. 3848–3852.

[12] O. Siméoni, C. Sekkat, G. Puy, A. Vobecky, É. Zablocki and P. Pérez, "Unsupervised object localization: Observing the background to discover objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Vancouver, BC, Canada, 2023, pp. 3176–3186.

[13] J. Philbin, J. Sivic, and A. Zisserman, "Geometric LDA: A generative model for particular object discovery," in *Proc. British Mach. Vis. Conf.*, Leeds, UK, 2008, pp. 1–10.

[14] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine, "Unsupervised object discovery: A comparison," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 284–302, Jun. 2010. doi: 10.1007/s11263-009-0271-8.

[15] O. J.´H´enaff *et al.*, "Object discovery and representation networks," in *Proc. Eur. Conf. Comput. Vis.*, Tel Aviv, Israel, 2022, pp. 123–143.

[16] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," 2021. doi: https://arxiv.org/pdf/2110.11334.

[17] X. Wang, Z. Zhu, C. Yao, and X. Bai, "Relaxed multiple-instance svm with application to object discovery," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 1224–1232.

[18] O. Simeoni, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Unsupervised object discovery for instance recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Lake Tahoe, NV, USA, 2018, pp. 1745–1754.

[19] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Adv. Space Res.*, vol. 41, no. 12, pp. 1955–1959, Jul. 2007. doi: 10.1016/j.asr.2007.07.020.

[20] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, San Francisco, CA, USA, 2010, pp. 1943–1950.

[21] A. Joulin, F. Bach, and J. Ponce, "Multi-class cosegmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Providence, RI, USA, 2012, pp. 542–549.
[22] R. Chen, L. Pan, C. Li, Y. Zhou, A. Chen and E. Beckman, "An improved deep fusion cnn for image recognition," *Comput. Mater. Contin.*, vol. 65, no. 2, pp. 1691–1706, Jun. 2020. doi: 10.32604/cmc.2020.011706.
[23] K. Tang, A. Joulin, L. J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, 2014, pp. 1464–1471.
[24] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Boston, MA, USA, 2015, pp. 1201–1210.
[25] X. Wang *et al.*, "FreeSOLO: Learning to segment objects without annotations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, New Orleans, LA, USA, 2022, pp. 14176–14186.
[26] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Generalized category discovery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, New Orleans, Louisiana, USA, 2022, pp. 7492–7501.
[27] Z. Lin, Z. Yang, and Y. Wang, "Foreground guidance and multi-layer feature fusion for unsupervised object discovery with transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, 2023, pp. 4043–4053.
[28] S. Kara, H. Ammar, F. Chabot, and Q. C. Pham, "Image segmentation based unsupervised multiple objects discovery," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 3276–3285.
[29] A. Dosovitskiy *et al.*, "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Rep.*, 2021, pp. 1–22. doi: 10.48550/arXiv.2010.11929.
[30] X. Wang, R. Girdhar, S. X. Yu, and I. Misra, "Cut and learn for unsupervised object detection and instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Vancouver, BC, Canada, 2023, pp. 3124–3134.
[31] C. Tang, L. Xie, X. Zhang, X. Hu, and Q. Tian, "Visual recognition by request," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Vancouver, BC, Canada, 2023, pp. 15265–15274.
[32] M. Wei, X. Yue, W. Zhang, S. Kong, X. Liu and J. Pang, "OV-PARTS: Towards open-vocabulary part segmentation," in *Neural Info. Processing Systems.*, New Orleans, Louisiana, USA, 2023, pp. 70094–70114.
[33] F. Locatello *et al.*, "Object-centric learning with slot attention," in *Neural Info. Process. Syst.*, 2020, pp. 11525–11538.
[34] G. Singh, Y. F. Wu, and S. Ahn, "Simple unsupervised object-centric learning for complex and naturalistic videos," in *Neural Info. Process. Syst.*, New Orleans, LA, USA, 2022, pp. 18181–18196.
[35] M. Seitzer *et al.*, "Bridging the gap to real-world object-centric learning," in *Proc. Int. Conf. Learn. Rep.*, Kigali, Rwanda, 2023, pp. 1–43. doi: 10.48550/arXiv.2209.14860.
[36] M. Caron *et al.*, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.
[37] B. Jia, Y. Liu, and S. Huang, "Improving object-centric learning with query optimization," in *Proc. Int. Conf. Learn. Rep.*, 2022, pp. 1–32.
[38] H. Zhang, Z. Wu, Z. Wang, Z. Chen, and Y. G. Jiang, "Prototypical residual networks for anomaly detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog.*, Vancouver, BC, Canada, 2023, pp. 16281–16291.
[39] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRAEM—A discriminatively trained reconstruction embedding for surface anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8330–8339.
[40] K. Perlin, "An image synthesizer," *ACM Siggraph Comput. Graph.*, vol. 19, no. 3, pp. 287–296, Jul. 1985.
[41] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. doi: 10.1109/34.868688.
[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, 2016, pp. 779–788.