MOST: Multiple Object localization with Self-supervised Transformers for object discovery

Sai Saketh Rambhatla[1,3*] rssaketh@meta.com
Ishan Misra[1] imisra@meta.com
Rama Chellappa[2,3] rchella4@jhu.edu
Abhinav Shrivastava[3] abhinav@cs.umd.edu

[1]Meta [2]Johns Hopkins University [3]University of Maryland, College Park

Abstract

*We tackle the challenging task of unsupervised object localization in this work. Recently, transformers trained with self-supervised learning have been shown to exhibit object localization properties without being trained for this task. In this work, we present **M**ultiple **O**bject localization with **S**elf-supervised **T**ransformers (MOST) that uses features of transformers trained using self-supervised learning to localize multiple objects in real world images. MOST analyzes the similarity maps of the features using box counting; a fractal analysis tool to identify tokens lying on foreground patches. The identified tokens are then clustered together, and tokens of each cluster are used to generate bounding boxes on foreground regions. Unlike recent state-of-the-art object localization methods, MOST can localize multiple objects per image and outperforms SOTA algorithms on several object localization and discovery benchmarks on PASCAL-VOC 07, 12 and COCO20k datasets. Additionally, we show that MOST can be used for self-supervised pretraining of object detectors, and yields consistent improvements on fully, semi-supervised object detection and unsupervised region proposal generation.Our project is publicly available at rssaketh.github.io/most.*

1. Introduction

Object detectors are important components of several computer vision systems such as visual relationship detection [21, 29], human-object interaction detection [1, 13, 45, 52], scene graph generation [57] and object tracking [54, 56] etc. Performance of object detectors is heavily reliant on the availability of training data. However, annotating large object detection datasets can be expensive and time consuming [14, 27]. Additionally, the vocabulary of object detectors is limited by the training datasets and such detectors often fail to generalize to new categories [7]. This strategy is not scalable and a more effective approach is warranted. Object discovery is one such task that has the potential to address these concerns. Object discovery is the problem of identifying and grouping objects/parts in a large collection of images without human intervention [23, 24, 36, 43]. The first step in object discovery is to obtain region proposals and subsequently group them semantically. Previous works on discovery used Selective Search [46], randomized Prim's [30] or a region proposal network (RPN) [35] to get object proposals. [48–50] used inter-image similarity to construct a graph and performed optimization or ranking, to localize objects without any supervision. Such methods are computationally expensive and often fail to scale to datasets larger than 20000 images. [34] used region proposals from an RPN and proposed a never ending learning approach and is the first method shown to scale to ∼100000 images. However, these region proposal methods are often of low quality, and therefore reduce the performance of discovery systems. Recently, LOST [41] and TokenCut [53] leveraged the object segmentation properties of transformers [47] trained using self-supervised learning (DINO [3]) to obtain high quality object proposals. They demonstrate significant improvements over

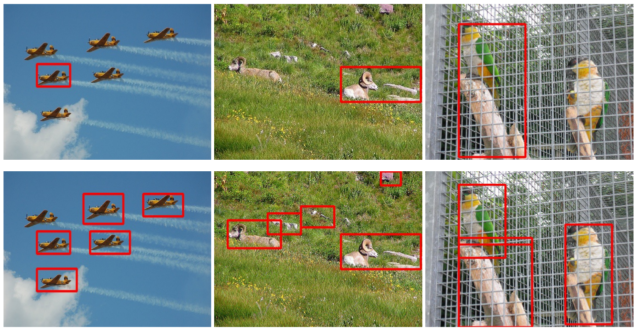


Figure 1: **Top**: Methods like LOST [41] (shown in figure), TokenCut [53] identify and localize the most salient foreground object and hence can detect only one object per image. **Bottom**: MOST is a simple, yet effective method that localizes multiple objects per image without training.

[*] Work done while at UMD.

state-of-the-art on object discovery, salient object detection and weakly supervised object localization benchmarks.

However, both LOST [41] and TokenCut [53] assume the presence of a single salient object per image and hence, can localize only one object as shown in Fig 1 (top). This assumption may hold for object centric datasets like ImageNet [37] but is not true for scene-centric real world datasets like PASCAL-VOC [12] and COCO [27]. In this work, we address the problem of localizing multiple objects per image and demonstrate the effectiveness of our approach for the task of unsupervised object localization and discovery on several standard benchmarks.

We propose a new object localization method called "Multiple Object localization with Self-supervised Transformers" (MOST) which is capable of localizing multiple objects per image without using any labels. We use the features extracted from a transformer [47] network trained with DINO [3]. Our method is based on two empirical observations; 1) Patches within foreground objects have higher correlation with each other than the ones on the background [41] and 2) The similarity map computed using the features of a foreground object with all the features in the image is usually more localized and less noisier than the one computed using the feature of a background. Our algorithm analyzes the similarities between patches exhaustively using a fractal analysis tool called box counting [28]. This analysis picks a set of patches that most likely lie on foreground objects. Next, we perform clustering on the patch locations to group patches belonging to a foreground object together. Each of these clusters is called *pools*. A binary mask is then computed for each *pool* and a bounding box is extracted. This capability enables the algorithm to extract multiple bounding boxes per image as shown in Fig.1 (bottom). We demonstrate that **without any training**, our method can outperform state-of-the-art object localization methods that train class agnostic detectors to detect multiple objects. To prove the effectiveness of MOST, we demonstrate results on several object localization and discovery benchmarks. On self-supervised pre-training for object detectors, using MOST yields consistent improvement across multiple downstream tasks using 6× fewer boxes. When compared against other self-supervised transformer-based localization methods, MOST achieves higher recall with and without additional training. We summarize the contributions of our work below.

- We propose MOST, an effective method to localize and discover multiple objects per image without supervision using transformers trained with DINO.

- We perform exhaustive experiments to assess the performance of MOST on several localization and discovery benchmarks and show significant improvements over the baselines.

The paper is organized as follows. In Section 2 we discuss related works on object localization and discovery. We describe our approach in detail in Section 3. We describe our experimental setup and present results in Section 4 and conclude in Section 5.

2. Related Works

Unsupervised Object Localization and Discovery: Object category discovery can be broadly segregated into image-based [15, 16, 18–20, 42] and region-based methods [4, 8, 22–24, 34, 41, 48–50, 53]. Region-based methods start by generating object proposals and later group them into coherent semantic groups. Image-based approaches on the other hand, assume the localization task to be solved and focus solely on the semantic grouping. Our current method is closer to the former. Vo *et al*., [48–50] localize regions in images by constructing an inter-image similarity graph between regions and partitioning it using optimization or ranking. Due to the quadratic nature of the graph, these methods cannot scale to large datasets beyond tens of thousands of images. Our current work does not compute inter-image similarity between regions and scales linearly with number of images. Lee *et al*., [24] propose object graphs that incorporates knowledge about a few known categories to facilitate the discovery of novel categories. MOST doesn't assume any prior knowledge and has the ability to discover objects from scratch. Lee *et al*., [23] extend object graphs to a curriculum based discovery pipeline, that learns to discover easy objects first and progressively proceeds to discover the harder ones. Along similar lines, Rambhatla *et al*., [34] propose a large scale discovery pipeline, similar to [24] that leverages prior knowledge about a few categories. Authors of [34] use an RPN [35] trained on known categories as the localization method. In contrast to that, MOST localizes objects in images using features of a transformer [47] trained with DINO [3].

Object localization using self-supervised networks: Recently, CNNs [17] and Transformers [47] trained in a self-supervised fashion, have been shown to exhibit object localization/segmentation properties [3, 10]. This property is of particular interest as it has the potential to facilitate research on unsupervised localization, detection and segmentation. [41] is a simple method, based on the observation that the key features of the last self attention layer of a transformer, trained using DINO, has certain affinity. They construct a map of inverse degree to extract bounding boxes on objects in an unsupervised fashion. This method is shown to outperform recent state-of-the-art methods by a significant margin. [53] propose an alternate method for localizing objects using self-supervised transformers, based on normalized cut [39]. TokenCut [53] construct an undirected graph based on token feature similarities and use normalized cut to cluster foreground and background patches. Spectral clus-

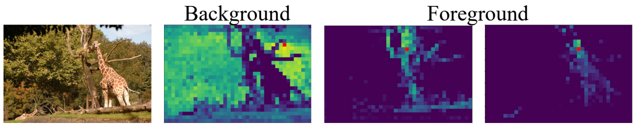


Figure 2: **Motivation for MOST**: Example showing similarity maps of tokens within background and foreground for an image from the COCO dataset. Similarity maps of tokens within foreground patches are less random spatially.

tering is used to solve the graph-cut and they show that the eigen vector corresponding to the second smallest eigenvalue provides a good cutting solution. TokenCut outperforms LOST on unsupervised object discovery. In addition to discovery, [53] also demonstrate impressive results on unsupervised saliency detection and weakly supervised object localization. Kyriazi *et al*., [31] propose deep spectral methods, that performs normalized cut [39] but using an affinity matrix computed using semantic and color features. Since this method is very similar to TokenCut and achieves lower performance, we only compare with the latter in this work.

However, one limitation of LOST and TokenCut is that they can localize only one object per image. Our proposed method, MOST can automatically localize multiple objects per image and outperforms LOST and TokenCut on standard discovery benchmarks.

3. Approach: MOST

MOST can be used to localize multiple objects in an image. Our approach, illustrated in Fig. 3, first identifies a set of tokens that is computed from patches on foreground objects. These tokens are then clustered and each cluster, named *pool*, is used to obtain a bounding box. Next, we discuss a few preliminaries in Section 3.1 followed by the motivation and proposed solution in Section 3.2.

3.1. Preliminaries

Box Counting: Box counting is a method of analyzing a pattern by breaking and analyzing it at smaller scales. This is done by performing a raster scan of the pattern at different scales and computing appropriate metrics to assess its fractal nature. In this work, we use a sliding window scan.

Vision Transformers: ViTs [9] operate on learned embeddings, called tokens, generated from non-overlapping image patches of resolution P×P (typically P=8/16) that form a sequence. To be precise, an image I of shape $H \times W \times 3$ is first divided into non-overlapping patches of resolution $P \times P \times 3$, generating a total of N = HW/P^2 patches. Next, each patch is embedded via a trainable linear layer to generate a token of dimension d to form a sequence of patches. An extra [CLS] token [6] is added to this sequence, whose purpose is to aggregate the information from the tokens of the sequence. Typically, a projection head is attached to the [CLS] to train the network for classification.

DINO: DINO [3] combines self-training and knowledge distillation without labels for self supervised learning. DINO constructs two global views and several local views of lower resolution, from an image. DINO consists of a teacher and a student network. The student processes all the crops while the teacher is operated only on the global crops. The teacher network then distills its dark knowledge to the student. This encourages the student network to learn local to global correspondences. In contrast to other knowledge distillation methods, DINO's teacher network is updated dynamically during training using exponential moving average. DINO was shown to perform on par or better than several baselines on several tasks of image retrieval, copy detection, instance segmentation etc. The property of importance to the current work, is the ability of DINO to localize foreground regions of semantic entities in an image. [41, 53] leverage this property to localize the salient object in an image by using the key features from the last self-attention layer. Similar to [41, 53], we concatenate the key features of all the heads in the last self-attention layer to obtain the input to MOST.

3.2. Multiple object localization

Motivation: Consider the example shown in Fig. 2. We show three examples of the similarity maps of a token (shown in red) picked on the background (column 2) and foreground (columns 3, 4). Tokens within foreground patches have higher correlation than the ones on background [41]. This results in the similarity maps of foreground patches being *less* "spatially" random than the ones on the background. The task then becomes to analyze the similarity maps and identify the ones with less spatial randomness. Box counting [25, 33] is a popular technique in fractal analysis that analyzes spatial patterns at different scales to extract desired properties. Hence, we adopt box counting for our case and since, we are interested in randomness, we adopt entropy as the metric.

Preprocessing: The input to our method is a d-dimensional feature $F \in \mathbb{R}^{N \times d}$, extracted from an image using a neural network. Here, N denotes the number of spatial locations in the feature map, in case of a CNN, or number of tokens, in case of a transformer network. The aim is to identify subsets of tokens, which we call *pools*, that can be used to localize all the objects in an image. We do not make any assumption on the number of objects present in the image. Given the feature F, we compute an outer product matrix $A = FF^T \in \mathbb{R}^{N \times N}$. Row i of matrix A, i.e., $A[i, :]$ encodes a similarity map of a token at location i with all the other tokens in F. Next, each row of A is processed by the Entropy-based Box Analysis (EBA) module.

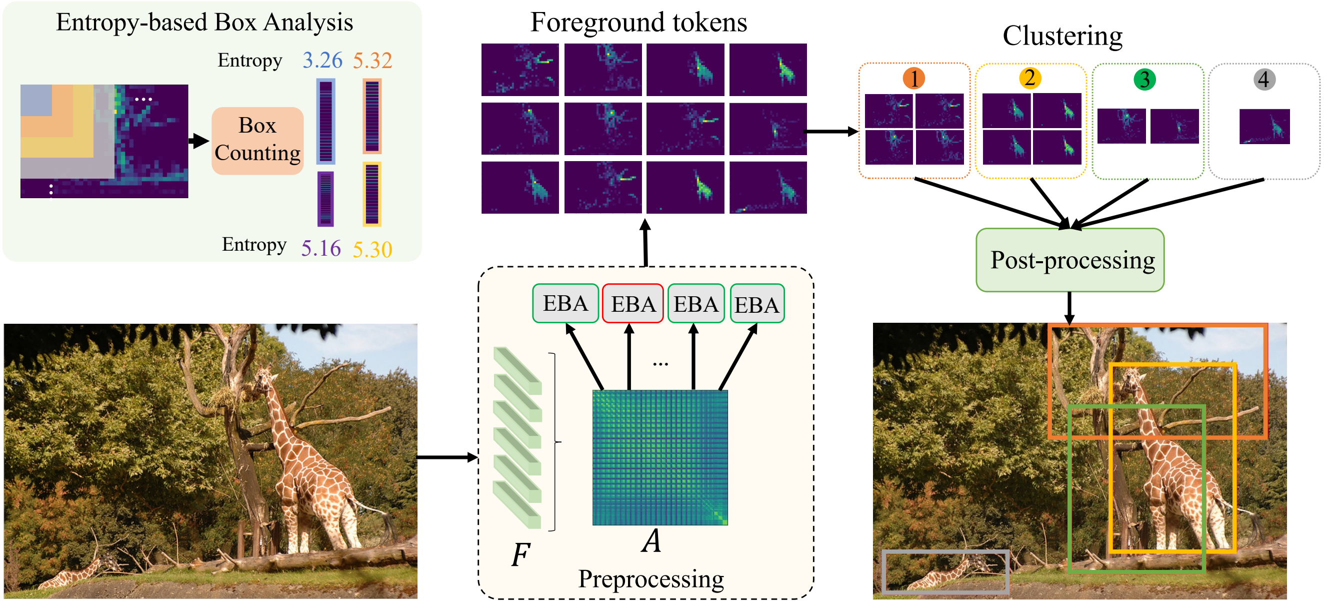


Figure 3: **Overview of MOST**: MOST operates on features extracted from transformers trained using DINO. The features are used to compute the outer product A. Each row of A is analyzed by the entropy-based box analysis (EBA) module that identifies tokens extracted from foreground patches. These patches are clustered using spatial locations as features to form *pools*. Each *pool* is then post-processed to generate a bounding box.

Entropy-based Box Analysis (EBA): The proposed entropy based box analysis module performs a fractal analysis method, called box counting to segregate similarity maps of tokens on foreground patches from those of background. As shown in Fig. 3, we perform a raster scan with increasing box (used interchangeably with kernel in this work) sizes on each map. Traditionally, measures like lacunarity [44] are computed within each box to analyze the pattern. In this work, we average the elements within each box. This can be implemented efficiently using pooling operations. The resulting downsampled map is flattened and the entropy is computed using the pmf computed as follows: $p(x = x_i) = \Sigma_{i=1}^{h.w} \frac{\mathbb{1}(f_i == x_i)}{h.w}$, where f_i is the i^{th} index in the feature map. A downsampled map belongs to a token on a foreground patch if its entropy is less than a threshold τ. Using K boxes in the EBA module results in K entropy values $e_k (k \in \{1, 2, \cdots, K\})$. Finally, we perform a majority voting among the entropies of all the downsampled maps, *i.e.*, $\Sigma_{i=1}^{K} \frac{\mathbb{1}(e_i \leq \tau)}{K} > 0.5$, to decide if the original similarity map belongs to a token on a foreground patch. A map of dimension $n \times n$ has a maximum entropy of $log(n^2)$. We use a threshold of the form $\tau = a + blog(n^2)$ (we use $a = 1, b = 0.5$ in this work). We do not consider τ as a hyperparameter and we pick a value that is mid-way between the minimum and maximum permissible value (b=0.5). To prevent a threshold of 0 for $n = 1$, we add a constant (a=1).

Clustering: The EBA module, identifies a set $\mathcal{S} = \{p | p \in \{1, 2, \cdots, N\}\}$, that contains the spatial locations of tokens computed from foreground patches. Often, highly redundant neighboring tokens are identified. We group neighboring tokens with the help of a clustering step to obtain *pools*. We convert the linear index p of the tokens to cartesian coordinates (x, y), and use that as the feature for clustering. Manhattan distance is used as the dissimilarity metric with a threshold ϵ ($\epsilon = 2$ i.e. Moore neighborhood). Since, the number of pools is not known a-priori, we use a density-based clustering method, DBSCAN [11] which automatically identifies the number of clusters from the data. *Pools* identified by the clustering step are then post-processed to obtain bounding boxes on foreground objects.

Post-processing: The clusters, called *pools*, obtained from the clustering step are then post-processed to obtain one box per *pool*. Consider M *pools* identified by the clustering step, i.e. $\mathcal{C}_i$, where $i \in \{1, 2, \cdots M\}$. Each *pool* $\mathcal{C}_i$ is a set of token locations $\mathcal{C}_i = \{p^i | p^i \in \{1, 2, \cdots, N\}\}$. We leverage the first observation mentioned above to obtain a bounding box from the *pool* as follows. First, we build a binary similarity matrix $\hat{A} = A > 0$. Next, within the tokens in the pool, we identify the one with lowest degree in $\hat{A}$, called the *core* token, c^*.

$$c^* = \arg\min_{c \in \mathcal{C}_i} d_c \quad \text{where} \quad d_c = \sum_{j=1}^{N} \hat{A}[c, j]$$

Authors of LOST [41] report that tokens with low degrees

Table 1: **Results on unsupervised pre-training of object detectors**. We train object detectors in a self-supervised fashion on the COCO dataset using different localization methods and compare their performance on the downstream tasks of semi and fully supervised object detection. COCO train set is used for fine-tuning and $k\%$ refers to the number of labeled samples used for training. Results are reported using AP[0.50:0.95] (denoted as AP), $AP_{0.50}$, and $AP_{0.75}$ on COCO validation set.

Method	Boxes per image	**VOC 07+12**						**COCO**					
		$k = 10\%$			fully supervised			$k = 1\%$	$k = 2\%$	$k = 5\%$	fully supervised		
		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}		AP		AP	AP_{50}	AP_{75}
LOST [41]	1	40.88	60.31	44.36	63.58	83.27	70.48	12.83 ± 0.32	17.23 ± 0.30	23.43 ± 0.38	44.30	62.80	48.40
TCut [53]	1	41.14	60.59	44.35	63.79	83.56	70.70	13.13 ± 0.38	17.27 ±0.21	23.27 ± 0.23	43.80	62.30	47.50
	5	39.12	57.51	42.29	63.44	83.14	70.35	13.57 ± 0.38	17.87 ± 0.32	23.17 ± 0.40	44.30	62.80	48.10
SS [46]	10	40.76	60.00	44.46	64.23	83.44	71.55	13.73 ± 0.29	18.00 ± 0.26	22.83 ± 0.25	43.90	62.60	47.60
	15	42.14	61.41	45.86	64.24	83.74	71.41	13.87 ± 0.29	18.23 ± 0.40	23.13 ± 0.11	44.30	62.60	48.30
MOST	4.65	43.03	63.29	46.61	64.34	84.12	71.77	13.93 ± 0.38	18.13 ± 0.25	22.63 ± 0.11	44.80	63.50	49.00
SS	30	42.12	61.20	45.71	64.84	83.98	71.76	14.47 ± 0.35	18.23 ± 0.42	**23.57 ± 0.21**	44.00	62.30	47.80
MOST	13.09	**44.40**	**63.83**	**48.28**	**65.24**	**84.24**	**72.37**	**14.83 ± 0.21**	**18.30 ± 0.17**	23.43 ± 0.45	**45.20**	**64.00**	**49.00**

most likely fall within an object. Next, we remove the tokens from the pool that do not correlate positively with c^* to form a reduced *pool* $\mathcal{C}_i^*$. This ensures that all the tokens in the current pool lie on the same foreground object. Next, a binary mask is constructed by computing the sum of similarities of token features in $\mathcal{C}_i^*$ with the features of all the tokens, i.e. $m_k^i = \mathbb{1}(\sum_{c \in \mathcal{C}_i^*} f_k^T f_c \geq 0)$. Finally, connected component analysis is performed on the binary mask and the bounding box of the island that contains c^* is selected as the region containing the object. We repeat this process for all the M *pools* to generate M bounding boxes per image. Note that, M is not assumed to be known a-priori and is decided automatically by our method. Additionally, we remove trivial boxes i.e., boxes which have area less than than a threshold (256) or cover the whole image.

Implementation Details: For all our experiments, we use the ViT-S/16 and ViT-B/8 [9] models trained with DINO [3] to extract the features. We concatenate the key features of all the heads from the last self-attention layer to use as the input to our method.

4. Experiments

In this section we describe, in detail, the experimental setup used for evaluation. We evaluate our method on two setups, namely the localization setup, and the discovery setup. We begin by describing the datasets and metrics in Sec. 4.1. We describe the evaluation setups in Sec. 4.2. Sec. 4.3 compares our method against contemporary work. We then describe ablation experiments in Sec. 4.4 and show qualitative results in Sec. 4.6.

4.1. Datasets and Metrics

We use the PASCAL-VOC [12] (2007, 2012 splits) and the COCO [27] (COCO20k [49] and COCO splits) datasets in our experiments. The PASCAL VOC [12] 2007 and 2012 trainval sets consists of 5011, 11540 images respectively, spanning twenty objects. The PASCAL VOC [12] test set consists of 4952 images. The COCO [27] 2014 train set consists of ∼110k images containing over eighty objects and the COCO minival set consists of 5000 images. We do not use any class or bounding box annotations for our method except for evaluation.

For the *localization* setup, we use the average precision at different thresholds ([0.5:0.95], 0.5 and 0.75), average recall (AR1, AR10 and AR100) and Correct Localization (CorLoc) metrics for evaluation. CorLoc is defined as the fraction of the images in which atleast one object is localized with an IoU greater than a threshold (0.5 in this work). AP, AR are defined in the usual way. For the object discovery setup, we report both the PASCAL VOC style AP_{50} and COCO style $AP_{[50:95]}$ metrics along with area under the purity-coverage plots [8, 34]. We refer the interested readers to [34] for definitions of purity and coverage.

4.2. Setups

Localization setup: This setup evaluates the localization performance of methods. We evaluate models on a) unsupervised pre-training, b) Multiple Object Localization, and c) single object localization. For unsupervised pre-training, localization methods are used to train object detectors in an unsupervised fashion and their performance is evaluated on the downstream task of object detection. In this work, we use the recently proposed DETReg [2] as the pre-training strategy which uses a Deformable DeTR [58] architecture. DETReg uses an object localization method and pre-trains an object detector in an unsupervised fashion. We evaluate the pre-trained model on the downstream tasks of semi-supervised, fully-supervised and class-agnostic object proposal generation. In the semi-supervised setting, models are trained on the PASCAL-VOC(07+12) and COCO train

Table 2: **Unsupervised class agnostic region proposal evaluation on COCO validation set**: We compare the performance of region proposals for training DETReg. Rk is Recall@k

Method	Boxes per image	AP	AP$_{50}$	AP$_{75}$	R1	R10	R100
LOST [41]	1	0.1	0.5	0	0.4	1.4	3.9
TCut [53]	1	0.3	1	0.1	**0.6**	**1.9**	**4.6**
	5	0.1	0.4	0	0.1	1	4.2
SS [46]	10	0.1	0.3	0	0.1	1.1	4.4
	15	0.1	0.3	0	0.1	1	4.1
	30	0.1	0.3	0	0.1	1	4
MOST	4.65	**0.8**	**1.4**	**1**	**0.6**	**1.9**	4.4

Table 3: **Results on object discovery**: Comparison of MOST with recent works on unsupervised object discovery. We experiment with three cluster numbers, *i.e.*, 20, 30, 40, on VOC 2007, 2007+12 and 80, 90, 100 on COCO20k.

Metric	Train →	**VOC 2007**			**VOC 07+12**			**COCO20k**		
	Clusters →	20	30	40	20	30	40	80	90	100
AP	LOST [41]	9.15	9.64	10.11	**10.95**	12.14	12.97	2.66	2.91	2.86
	MOST	**9.20**	**10.07**	**11.09**	10.12	**12.89**	**13.30**	**3.13**	**3.18**	**3.32**
AP$_{50}$	LOST [41]	**26.32**	27.78	29.46	**29.35**	33.27	**34.80**	7.17	7.72	7.87
	MOST	25.35	**28.19**	**31.31**	27.04	**34.40**	34.54	**8.13**	**8.14**	**8.76**

Table 4: **Results on single-object localization**: Comparison of MOST with recent object discovery methods on VOC 07, 12 and COCO20k using CorLoc.

Method	**VOC 07**	**VOC 12**	**COCO20k**
rOSD [49]	54.5	55.3	48.5
LOD [50]	53.6	55.1	48.5
DINO-seg† [3]	45.8	46.2	42.1
LOST [41]	61.9	64.0	50.7
TokenCut [53]	68.8	72.1	58.8
LOST [41] + CAD	65.7	70.4	57.5
TCut [53]+CAD	71.4	75.3	62.6
MOST	**74.8**	**77.4**	**67.1**

sets without labels and are fine-tuned on $k\%$ of labeled data similar to [2]. In the fully supervised setting, pre-trained models are fine-tuned on the full PASCAL-VOC and COCO dataset using all the labels. For the class-agnostic object proposal generation, models are pre-trained on COCO dataset without labels and the generated object proposals are evaluated on the COCO validation set similar to [2].

We follow the settings used in [50] for multiple-object localization and evaluate on PASCAL-VOC 2007 and COCO20k. For single-object localization, we follow the settings in [41, 53] and evaluate on PASCAL-VOC 2007, PASCAL-VOC 2012 and COCO20k.

Discovery setup: This setup evaluates the object discovery performance. Similar to [41] we use the regions obtained by our localization method, to perform K-means clustering and use the resulting cluster labels to train Faster-RCNN object detectors on PASCAL-VOC 2007, 2012 trainval and COCO20k train sets. We report results of these experiments on the PASCAL-VOC 2007 test and COCO minival sets respectively. In addition to this, we report the performance of our discovery method on COCO train set, similar to the large scale discovery in [34].

4.3. Comparison with contemporary methods

In this section we compare our method against contemporary works the *localization* and *discovery* setups.

4.3.1 Localization setup

Unsupervised Pre-training: Table 1 compares the results of all the localization methods on unsupervised pre-training of object detectors. We use average precision at different IoU thresholds ([0.50:0.95]: AP, 0.5: AP$_{0.50}$, 0.75: AP$_{0.75}$) for evaluation. On the semi-supervised setting, on VOC 07+12 ($k = 10\%$), the self-supervised transformer based methods (LOST, TokenCut and MOST) outperform SS [46] with fewer boxes per image. In particular, TokenCut (denoted as TCut in Table 1) which outputs only one box per image, outperforms SS, using ten boxes per image, by $\sim$0.4 points on mAP. MOST which outputs an average of 4.65 boxes per image outperforms TokenCut (the best performing self-supervised transformer based method) by 1.89, 2.7 and 2.26 percentage points on AP, AP$_{50}$, and AP$_{75}$ respectively. This can be attributed to the ability of MOST to output multiple foreground regions resulting in more samples for pre-training which is not possible in the case of TokenCut. MOST outperforms SS, that outputs 30 boxes per image, by 0.91, 2.09 and 0.9 points on AP, AP$_{50}$, and AP$_{75}$ respectively using almost $6\times$ fewer boxes per image and this can be attributed to the ability of MOST to generate high quality proposals. On COCO, MOST outperforms TokenCut by 0.8 and 0.86 on the 1% and 2% setting of semi-supervised learning. MOST with a ViT-B/8 backbone, that outputs 13.09 boxes on average per image, outperforms SS (with 30 boxes per image) by 0.36, 0.07 points on k=1% and k=2% few shot splits of COCO respectively.

On the fully supervised setting, MOST outperforms LOST and TokenCut by 0.76 and 0.55 (AP) percentage points respectively on VOC 07+12. On COCO, MOST outperforms them by 0.50 and 1 points respectively. On VOC 07+12, MOST using ViT-B/8 (13.09 boxes per image) outperforms SS (with 30 boxes per image) by 0.40. On the much harder COCO dataset, MOST outperforms SS 1.20 (AP) percentage points using $2\times$ fewer boxes per image.

In Table 2 we report the class agnostic object proposal evaluation of DETReg trained using different localization methods. We report average precision at different IoU thresholds (AP, AP$_{50}$, AP$_{75}$) and recall @ 1, 10 and 100 proposals per image (denoted as R1, R10, and R100) to evaluate the quality of region proposals. Note that the numbers in the table are low because of the unsupervised nature of training. All the self-supervised transformer-based methods achieve performance better than SS with far fewer boxes. In recall, TokenCut and MOST perform on par with each other and outperform rest of the methods with significant improvements. MOST achieves the highest performance on average precision among all the methods. It can achieve

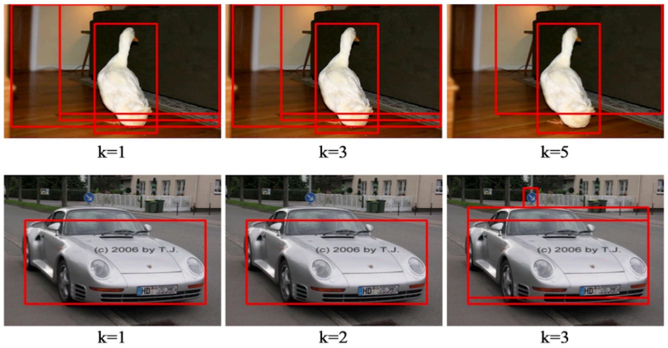


Figure 4: **Effect of kernel size**: Different kernel sizes can identify different tokens as belonging to the foreground. Multiple kernels help eliminate noisy predictions (first triplet) and missed predictions (second triplet).

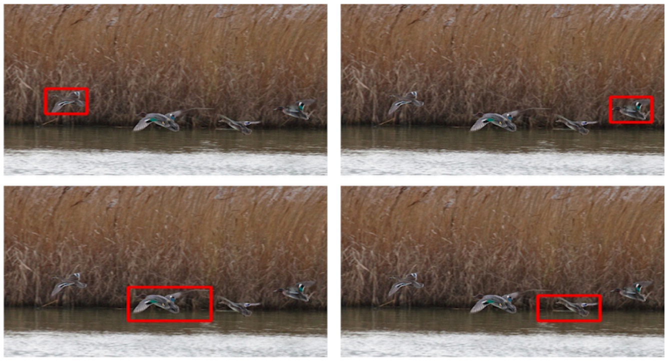


Figure 5: Figure demonstrating the effect of clustering in MOST: Each image consists of a bounding box generated from a *pool*. We observe that each *pool* focuses on different foreground instance.

higher precision and recall because of its ability to output multiple high quality regions per image. While LOST and TokenCut output high quality boxes, they cannot output more than one box per image. SS on the other hand, outputs multiple boxes but with poor quality.

Multiple Object Localization: We compare with LOD, the state-of-the-art method on the multi-object localization benchmark proposed by LOST using the code released by authors. On VOC2007, we attain an odAP[0.5:0.95] of 6.43 compared to 5.35 attained by LOD, an improvement of 1.09 percentage points. On the COCO20k dataset, we attain a performance of 1.70 (compared to 1.53 achieved by LOD) on the harder odAP[0.50:0.95] metric. Note, we do not compare with rOSD [49] as LOD [50] outperforms it.

Single Object Localization: Table 4 compares the results of our method on single object localization with recent methods on PASCAL VOC 2007, 2012 and COCO20k respectively. We use the CorLoc metric to evaluate methods. Note that MOST is a multiple object localization method and this setup evaluates the ability of methods to output a single region. Since MOST outputs multiple boxes, we use the heuristic, average best overlap (for evaluating object proposals in [46]), to select one region per image. The numbers reported for MOST in this table are the "best" case scenario. We outperform LOST by 12.9, 13.4 and 16.4 percentage points on VOC 2007, 2012 and COCO20k respectively. We outperform TokenCut [53] by 6, 5.3 and 8.3 percentage points on the three datasets respectively. To obtain multiple regions per image, authors of LOST train a foreground object detector using the regions obtained by their method as supervision, called LOST+CAD [41]. This method can output multiple boxes per image and from Table 4, even without any training, our method outperforms LOST+CAD and TokenCut+CAD by 9.1, 7, 9.6 and 3.4, 2.1, 4.5 percentage points on VOC 2007, 2012 and COCO20k respectively.

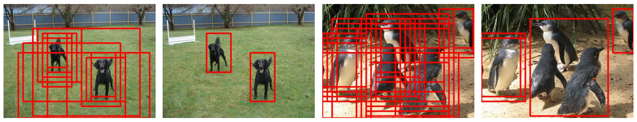


Figure 6: Figure demonstrating the effect of clustering in MOST. Eliminating the clustering results in noisier outputs.

Discovery Setup: This setup evaluates the true object discovery performance as the localized boxes are used to discover semantic groups. Following LOST [41], we first cluster the features of the localized objects using K-means clustering. For VOC 2007 and 2007+2012 trainval splits, we use 20, 30 and 40 clusters. We use 80, 90 and 100 clusters for COCO20k train split. We report the results of experiments on VOC 2007, 07+12 trainval sets on VOC 2007 test set. For experiments on COCO20k, we report results on the COCO validation set. Results are tabulated in Table 3. MOST outperforms LOST in most settings with the margin of improvement higher for more number of clusters and more cluttered datasets like COCO. For more details on clustering and training refer to supplementary.

Finally, we evaluate the performance of MOST on large-scale object discovery setup introduced in [34]. For this setup, we use the area under the purity coverage plot as the metric. [34] automatically identifies the number of clusters and obtains an AuC@0.5 of 3.6% on the COCO 2014 train set. We extract the DINO [CLS] token features of regions obtained from MOST for K-Means clustering. To avoid specifying the number of clusters manually, we employ the "kneedle" method [38] to get the optimal number of clusters (more details in supplementary). Next, we randomly sample 10000 features from the whole dataset and cluster them using K-means with the optimal number of clusters. This subsampling avoids loading all the features into mem-

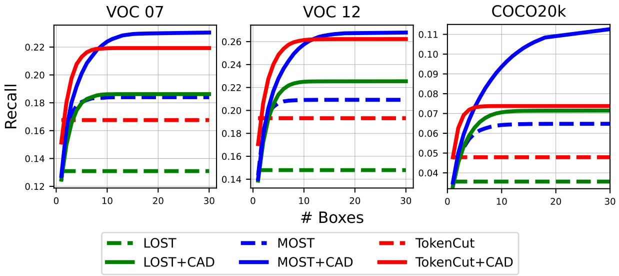


Figure 7: **Recall analysis**: Comparison of recall values of MOST, MOST+CAD with LOST and LOST+CAD. LOST generates one bounding box per image. MOST+CAD, MOST have higher recall and cover more ground-truth objects for a fixed set of boxes.

ory. MOST + optimal K-means achieves an AuC@0.5 of 8.74% on COCO 2014 train set. We use the cluster labels to train an object detector on the COCO train set and achieve an AP/AP$_{50}$ of 3.9/9.5% compared to 5.2% AP$_{50}$ obtained by [34] on COCO validation set. For more experiments on unsupervised saliency detection and weakly supervised localization, refer to the supplementary.

4.4. Ablation Experiments

Effect of kernel size: The EBA module performs box analysis in a sliding window fashion using boxes (or kernels) of different sizes. We implement this efficiently using a pooling operation. We visualize the effect of the size of pooling kernels on the final output in Fig. 4. We observe that the majority voting performed in EBA, helps in removing noisy predictions in the first triplet, where a box identified by kernel of size 1 is eliminated by majority voting of kernels with larger receptive field. In the second triplet in Fig. 4, an object which was missed by the lower order kernels (k=[1,4]), can be picked up with a higher order kernel (k=5). For all our experiments, we use kernels of sizes upto 5 (*i.e.* k=[1,5]). For quantitative results on the effect of kernel sizes, refer to the supplementary.

Effect of clustering: MOST performs clustering with the token locations as features to obtain *pools*. Each pool contains tokens belonging to a foreground object. We show the effect of clustering qualitatively in Fig. 5. We observe that each pool focuses on one foreground object and illustrate the bounding boxes extracted from each *pool*.

Each pool contains tokens belonging to a foreground object. We show the effect of forming *pools* qualitatively in Fig. 6. The first image in each pair shows the output of MOST without clustering to form *pools* and the second image shows the output with the formation. We observe that, without clustering, each token can generate a bounding box resulting in noisier outputs.

Recall of boxes: To analyze the object localization performance of MOST, we compare its recall with LOST and TokenCut on VOC 07, 12 and COCO20k datasets in Fig. 7. The x-axis represents the maximum number of boxes allowed per image and the y axis plots the recall. LOST and TokenCut generate only one box per image and hence have fixed recall in all the plots. MOST can generate more boxes and hence have higher recall than LOST and TokenCut. [41] trains a class agnostic detector (CAD) to output multiple boxes per image using the output of LOST as supervision. Without a single step of training, MOST performs competitively against LOST+CAD on all the datasets. With a class agnostic detector, MOST+CAD outperforms LOST, TokenCut and their CAD counterparts comfortably on all the datasets. On COCO20k, a much harder dataset, MOST+CAD outperforms all the methods with a significant margin demonstrating its superior localization abilities.

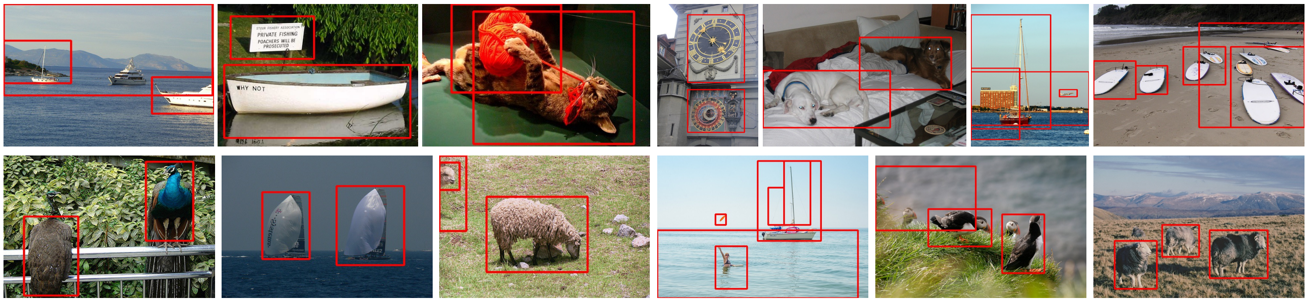
Table 5: Effect of backbone on single-object localization

Backbone	VOC 2007	VOC 2012	COCO20k
ViT-S/8	84.25	86.00	80.50
ViT-S/16	74.80	77.40	68.60
ViT-B/8	85.40	87.00	81.73
ViT-B/16	72.72	76.28	67.20

Effect of EBA: We study the effect of EBA on single-object localization. The task of the EBA module is to identify tokens on foreground instances from similarity maps. We replace the EBA module with the strategy used by LOST [41], effectively giving LOST the ability to localize multiple objects. We use top-100 patches and this system achieves a CorLoc of 63.66 (compared to 74.84 of MOST). The EBA module can automatically pick the right tokens, unlike LOST to localize multiple objects. This experiment demonstrates the benefit of the proposed EBA module.

Timing analysis: MOST can localize multiple objects per image and does so by analyzing the similarity maps of all the tokens which is computationally more expensive. To understand this requirement, we perform a timing analysis. On average, LOST takes 0.008s per image while MOST takes 0.3s. MOST obtains a recall of 0.19, 0.21, and 0.08 on VOC07, VOC12, and COCO20k respectively (compared to 0.13, 0.15, and 0.03 of LOST). This translates to localizing an additional 750, 2400, and 7200 instances than LOST on VOC07, VOC12, and COCO20k respectively. We believe the additional time taken by MOST is justified by the improvement in recall performance which is essential for an object localization method.

Effect of backbone: We study the effect of backbone on MOST in Table 5. We observe that backbones with a smaller patch size can localize more objects, especially smaller ones (refer to supplementary for qualitative results)



(a) VOC 07+12

(b) COCO

Figure 8: **Qualitative results of MOST on VOC 07, 12 and COCO**: MOST can localize multiple objects in cluttered scenes without training. Localization ability of MOST is not limited by the biases of annotators and can localize rocks, mountains, branches, water bodies, *etc*.

Table 6: **Results on unsupervised saliency detection**: We compare MOST to state-of-the-art unsupervised saliency detections methods. Top results are in bold and second best is highlighted in blue. Ability to detect multiple objects in images is a good tradeoff for a slight drop in performance on saliency detection

Method	**ECSSD**			**DUTS**			**DUT-OMRON**		
	max F_β	IoU	Acc (%)	max F_β	IoU	Acc (%)	max F_β	IoU	Acc (%)
DeepUSPS[32]	58.4	44.0	79.5	42.5	30.5	77.3	41.4	30.5	77.9
BigBiGAN [51]	78.2	67.2	89.9	60.8	49.8	87.8	54.9	45.3	85.6
E-BigBiGAN [51]	79.7	68.4	90.6	62.4	51.1	88.2	56.3	46.4	86.0
LOST [41]	75.7	62.5	88.0	61.8	52.0	87.1	47.4	40.2	79.6
TCut [53]	**80.3**	**71.2**	**91.8**	**67.2**	**57.6**	**90.3**	**60.0**	**53.3**	**88.0**
MOST	79.1	63.1	89.0	66.6	53.8	89.7	57.0	47.5	87.0

resulting in a higher CorLoc. This comes at a cost of producing noisier outputs. We refer interested readers to the supplementary for more analysis and qualitative results on the effect of backbones and patch sizes.

4.5. Additional Experiments

Unsupervised Saliency Detection: MOST can easily be extended to unsupervised saliency detection. We experiment with ECSSD [40], DUTS [26] and DUT-OMRON [5] datasets and use the metrics used by [41, 53] for evaluation. All these datasets require methods to segment the salient object in an image. Hence, naively using MOST doesn't perform well. To extend MOST for saliency detection, we select the largest *pool* and use the similarity map computed using its tokens as the saliency map. In Table 6, we compare MOST with TokenCut and LOST on the three datasets. MOST outperforms LOST comfortably on all the metrics, datasets and fares competitively against TokenCut. We believe that the ability to detect multiple objects in images is a good tradeoff for a slight drop in performance on saliency detection. We refer interested readers to the supplementary for qualitative results on saliency detection.

Weakly supervised localization: We evaluate MOST on weakly supervised object localization on CUB-200 [55] and Imagenet [37] datasets respectively, and achieve a CorLoc of 92.42 (*vs*. 91.8 of TokenCut) on CUB-200 and a CorLoc of 71.4 (*vs*. 65.4 of TokenCut).

4.6. Qualitative Results:

We illustrate qualitative results of MOST on VOC2007, 2012 and COCO datasets in Fig. 8. Fig. 8a shows results on VOC 2007 and 2012 and Fig. 8b shows results on COCO dataset. MOST is capable of localizing fairly complex scenes in all the three datasets. We observe that, such unsupervised localization methods are not limited by the categories annotated by humans but can localize regions of "stuff" like sign boards (third image in the first row of Fig. 8a right), rocks (last image of last row in Fig. 8b), and water bodies (first image in second row of Fig. 8b).

5. Conclusion

We present MOST, an effective method for localizing multiple objects in complex images without a single annotation. MOST leverages object segmentation properties of transformers trained using DINO [3]. We show that the ability of MOST to localize multiple objects in an image is very effective on several object localization and discovery benchmarks. In particular, MOST outperforms recent state-of-the-art methods that train a class agnostic detector, on the task of single object localization, without any training. Further, we show that MOST achieves higher recall and covers more ground truth objects for a fixed set of boxes than LOST [41], a contemporary work on object localization. Finally, we extend MOST to the task of unsupervised saliency detection and report competitive results with recent works.

Acknowledgement. This project was partially supported by DARPA SemaFor (HR001119S0085) and DARPA SAIL-ON (W911NF2020009), and Amazon Research Award to Abhinav Shrivastava. Rama Chellappa was supported by an ONR MURI (N00014-20-1-2787).

References

[1] Ankan Bansal, Sai Saketh Rambhatla, Abhinav Shrivastava, and Rama Chellappa. Detecting human-object interactions via functional generalization. *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020. 1

[2] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection, 2021. 5, 6

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 3, 5, 6, 9

[4] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised Object Discovery and Localization in the Wild: Part-based Matching with Bottom-up Region Proposals. In *CVPR - IEEE Conference on Computer Vision & Pattern Recognition*, pages 1201–1210, Boston, United States, June 2015. IEEE. 2

[5] Yang Chuan, Zhang Lihe, Lu Huchuan Ruan Xiang, and Yang Ming-Hsuan. Saliency detection via graph-based manifold ranking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3166–3173. IEEE, 2013. 9

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 3

[7] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 1

[8] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Context as supervisory signal: Discovering objects with predictable context. In *European Conference on Computer Vision (ECCV)*, pages 362–377. Springer, 2014. 2, 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3, 5

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9588–9597, October 2021. 2

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996. 4

[12] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 2, 5

[13] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8359–8367, 2018. 1

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1

[15] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[16] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 2

[17] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[18] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Deep image category discovery using a transferred similarity function. *CoRR*, abs/1612.01253, 2016. 2

[19] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[20] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2

[21] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2016. 1

[22] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised Object Discovery and Tracking in Video Collections. In *ICCV - IEEE International Conference on Computer Vision*, pages 3173–3181, Santiago, Chile, Dec. 2015. IEEE. 2

[23] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2010. 1, 2

[24] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, pages 1721–1728. IEEE, 2011. 1, 2

[25] Jian Li, Qian Du, and Caixin Sun. An improved box-counting method for image fractal dimension estimation.

Pattern Recognit., 42:2460–2469, 2009. 3

[26] Wang Lijun, Lu Huchuan, Wang Yifan, Feng Mengyang, Wang Dong, Yin Baocai, and Ruan Xiang. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 9

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 1, 2, 5

[28] Jing Z. Liu, Luduan Zhang, and Guang H. Yue. Fractal dimension in human cerebellum measured by magnetic resonance imaging. *Biophysical journal*, 85 6:4041–6, 2003. 2

[29] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *European Conference on Computer Vision*, 2016. 1

[30] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim's algorithm. In *2013 IEEE International Conference on Computer Vision*, pages 2536–2543, 2013. 1

[31] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[32] Duc Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Thi-Phuong-Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction with self-supervision. In *NeurIPS*, 2019. 9

[33] Roy E Plotnick, Robert H Gardner, William W Hargrove, Karen Prestegaard, and Martin Perlmutter. Lacunarity analysis: a general technique for the analysis of spatial patterns. *Physical review E*, 53(5):5461, 1996. 3

[34] Sai Saketh Rambhatla, Ramalingam Chellappa, and Abhinav Shrivastava. The pursuit of knowledge: Discovering and localizing novel categories using dual memory. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9133–9143, 2021. 1, 2, 5, 6, 7, 8

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 1, 2

[36] Alvaro Collet Romea, Bo Xiong, Corina Gurau, Martial Hebert, and Siddhartha Srinivasa. Herbdisc: Towards lifelong robotic object discovery. In *Proceedings of International Journal of Robotics Research (IJRR)*, January 2014. 1

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2, 9

[38] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, 2011. 7

[39] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 731–737, 1997. 2, 3

[40] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:717–729, 2016. 9

[41] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *Proceedings of the British Machine Vision Conference (BMVC)*, November 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9

[42] Krishna Kumar Singh, Utkarsh Ojha, and Yong Jae Lee. Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*, 2019. 2

[43] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 370–377 Vol. 1, Oct 2005. 1

[44] T. G. Smith, G. D. Lange, and W. B. Marks. Fractal methods and results in cellular morphology — dimensions, lacunarity and multifractals, Nov. 1996. 4

[45] Masato Tamura, Hiroki Ohashi, and Tomoaki Yoshinaga. Qpic: Query-based pairwise human-object interaction detection with image-wide contextual information. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10405–10414, 2021. 1

[46] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. 1, 5, 6, 7

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 2

[48] Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8287–8296, 2019. 1, 2

[49] Huy V. Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5, 6, 7

[50] Huy V. Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2021. 1, 2, 6, 7

[51] Andrey Voynov, Stanislav Morozov, and Artem Babenko.

Object segmentation without labels with large-scale generative models. In *ICML*, 2021. 9

[52] Tiancai Wang, Tong Yang, Martin Danelljan, Fahad Shahbaz Khan, X. Zhang, and Jian Sun. Learning human-object interaction detection using interaction points. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4115–4124, 2020. 1

[53] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 5, 6, 7, 9

[54] Zhongdao Wang, Liang Zheng, Yixuan Liu, and Shengjin Wang. Towards real-time multi-object tracking. *The European Conference on Computer Vision (ECCV)*, 2020. 1

[55] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. 9

[56] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE, 2017. 1

[57] Danfei Xu, Yuke Zhu, Christopher Bongsoo Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3106, 2017. 1

[58] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 5