CuVLER: Enhanced Unsupervised Object Discoveries through Exhaustive Self-Supervised Transformers

Shahaf Arica Or Rubin Sapir Gershov Shlomi Laufer
Technion - Israel Institute of Technology
shahaftech@gmail.com

Abstract

In this paper, we introduce VoteCut, an innovative method for unsupervised object discovery that leverages feature representations from multiple self-supervised models. VoteCut employs normalized-cut based graph partitioning, clustering and a pixel voting approach. Additionally, We present CuVLER (Cut-Vote-and-LEaRn), a zero-shot model, trained using pseudo-labels, generated by VoteCut, and a novel soft target loss to refine segmentation accuracy. Through rigorous evaluations across multiple datasets and several unsupervised setups, our methods demonstrate significant improvements in comparison to previous state-of-the-art models. Our ablation studies further highlight the contributions of each component, revealing the robustness and efficacy of our approach. Collectively, VoteCut and CuVLER pave the way for future advancements in image segmentation. The project code is available on GitHub at https://github.com/shahaf-arica/CuVLER

1. Introduction

Object localization remains a cornerstone in computer vision, empowering AI systems with abilities ranging from perception and inference to strategic planning and interactions centered around objects. The conventional training paradigm for such models often requires specialized annotations—be it object bounding boxes, masks, or localized keypoints. Unfortunately, acquiring these manual annotations is time-consuming and resource-heavy [38]. Consequently, there is a burgeoning interest in automated object detection and segmentation, particularly in unsupervised settings, circumventing the exhaustive annotation process [39].

Wang *et al*. [33] introduced CutLER (Cut-and-LEaRn), a new approach for training unsupervised object detection and segmentation models. CutLER employs a two-step process. First, it generates pseudo-labels using the MaskCut method. This novel approach leverages a *single* self-supervised model to create a *fixed* number of pseudo-labels per image. In the second phase, these pseudo-labels train a segmentation model, resulting in the CutLER model. Additionally, CutLER exhibits potential as a base model for supervised detection, demonstrating efficacy in few-shot benchmarks.

Our research builds on this pioneering work, progressing through three critical stages: (1) We employ our innovative method, referred to as 'VoteCut', to harness feature representations from *multiple* Vision Transformers (ViTs) [14] trained in a self-supervised manner [8, 27] to generate pseudo-labels with corresponding confidence scores, aided by the eigenvectors of Normalized Cuts (NCut) [28]; (2) The generated pseudo-labels are then used to train a robust object detector, which we refer to as "CuVLER"; (3) The output from this detector aids mask refinement in a separate domain, involving the generation of pseudo-labels from detector predictions in this new domain, followed by filtering. These refined pseudo-labels undergo a subsequent retraining phase, encapsulating our self-training approach.
Our methodology, the proposed CuVLER model, distinguishes itself from Wang *et al*.'s work in its capability to generate superior object masks and detections in a given domain, without the need of several "in domain" self-training stages. We achieve this by integrating insights from *multiple* ViT models, enhancing the cluster separation process. We also pioneer a self-training strategy that operates within the original and the target domain, unlike the approach in CutLER, which restricts self-training to the original domain. This innovation allows our model to enter self-training stages beyond its initial domain, achieving significant enhancements after just one epoch, showcasing its efficiency and flexibility.

This paper underscores the following contributions in the domain of unsupervised object detection and segmentation:

1. **In-Domain Mask and Detection Discovery - VoteCut**: At the heart of our work is a novel method for identifying high-quality masks and detections within a specific domain. We considerably elevate object localization and segmentation's efficacy by harnessing *multiple*

self-supervised models, paving the way for future explorations. Notably, unlike MaskCut [33], these masks are equipped with a confidence score, enhancing their reliability and utility.

2. **Instance-Level Loss Function with Soft-Targets**: We present a unique loss function that operates at the instance level and integrates soft-targets. This innovation facilitates a more granular training regimen, boosting object segmentation and detection.
3. **Cross-Domain Learning via Self-Training**: Highlighting our method's adaptability, we delineate how our distinctive loss function can be harnessed both within and outside its original domain in a self-training context. Such versatility underscores our model's potential to be adapted across diverse applications, enriching the unsupervised object detection and segmentation landscape.

2. Related work

Self-supervised feature learning. Self-supervised learning aims to generate rich data representations without reliance on human annotations, typically achieved through pretext tasks. A noteworthy advancement in this domain has been the training of Vision Transformers (ViTs) [14] in a self-supervised manner, which yields high-quality features. Broadly, pretext tasks fall into two categories: Augmentation-based and Reconstruction-based. Augmentation-based methods posit that varying augmentations of a single sample should produce semantically similar outcomes compared to disparate dataset samples. This often takes the shape of distinguishing augmentations from unrelated samples via constructive [9, 20, 26, 36], similarity [10, 18], clustering [2, 7, 37], or category-based [8, 27] feature learning. Reconstruction-based methods, on the other hand, emphasize reconstructing hidden patches or pixels, aiming to discern object structures within the image [3, 13, 21].

Unsupervised instance segmentation. Attaining unsupervised instance segmentation, as demonstrated by FreeSOLO [32], involves the extraction of preliminary coarse object masks, followed by mask refinement through a self-training procedure. While FreeSOLO can generate multiple coarse masks per image, their quality occasionally falls short. Similar to our approach, other techniques harnessed DINO [8]-extracted features to pursue instance segmentation. These endeavors are motivated by the observation that DINO features encapsulate meaningful interconnections between patches within each image. Models like LOST[29] and TokenCut[34] leverage self-supervised ViT features for segment discovery through a graph constructed from patch key features' similarity matrix connections. However, their emphasis often remains restricted to the image's primary salient object. Conversely, MaskDistill [31] extracts class-agnostic initial masks from a self-supervised DINO's affinity graph, yet its single-mask approach during distillation significantly restricts multi-object detection. CutLER [33] has indeed carved a significant mark in object detection and segmentation; however, our method's novelty stems from leveraging multiple models and achieving exceptional results without the need for extensive "in domain" self-training stages, making it a promising advancement in the field of unsupervised object detection and segmentation.

3. Method

This study introduces an innovative approach for unsupervised object detection and segmentation using the "cut-vote-and-learn" pipeline. This technique capitalizes on the findings from recent research [29, 33, 34] highlighting the effectiveness of self-supervised representations for object discovery. Our pipeline, illustrated in Figure 1, presents a straightforward technique capable of detecting multiple objects, resulting in substantial improvements in segmentation and detection performance within the target domain. Specifically, we first introduce VoteCut, which generates multiple binary masks per image using self-supervised features from DINO [8] in the ImageNet domain [11]. We then leverage a loss function with soft targets to enable self-training with these masks. From this point forward, VoteCut with an additional self-training process using our novel loss function will be called "CuVLER". Additionally, we also enhanced CuVLER performance through self-training across different domains.

3.1. Normalized Cuts

Normalized Cuts (NCuts) [28] is a popular algorithm for image segmentation and clustering. According to this technique, we represent each patch as a node, thus constructing an undirected, fully connected graph. Each pair of nodes within this graph is connected by a weighted edge that measures their similarity. The NCut algorithm attempts to minimize the cost of partitioning the aforementioned graph into sub-graphs by solving a generalized eigenvalue problem:

$$(D - W)x = \lambda Dx \tag{1}$$

Where W and D denote the adjacency matrix and the degree matrix of the weighted graph, respectively [5]. The solution denoted as x in Eq. (1), corresponds to the eigenvector associated with the second smallest eigenvalue λ. Traditionally, the formulation fixes the number of clusters in the graph, usually a bipartition; however, we preferred a more relaxed approach that utilizes the K-means algorithm [12, 30].

3.2. VoteCut for object discoveries

This work presents a novel technique designed to harness the collective power of multiple self-trained models via a

voting mechanism, leading to precise object segmentation (see Figure 1b). We capture diverse image content perspectives using an ensemble of models trained on different augmented image sets. With their varying transformer patch sizes, these models can focus on distinct image attributes, thus maximizing object detection precision. Our proposed method maximizes the collective intelligence of the aforementioned models by conducting a voting procedure on each image segment. It prioritizes the most widely agreed-upon masks while diminishing the influence of masks with fewer votes.

In line with the methodology presented in TokenCut by Wang et al. [34], we adopt a procedure involving the extraction of the second smallest eigenvector from each model, as determined by the Normalized Cut (NCut) algorithm [28], for each input image.

In this study, we employ this approach on multiple DINO and DINOv2 [27] models, which exhibit different patch sizes, to obtain feature representations for individual patches. Subsequently, these representations are used to construct the affinity matrix employed in the NCut algorithm. To calculate elements in the affinity matrix W, we use the cosine similarity between the patch's features. For DINO models, we follow Wang *et al*. [33] and use the 'key' features extracted from the endmost attention layer. When using a DINOv2 model, the features used instead are the output features of the endmost attention layer. The calculation is detailed in Equation (2), where K_i is the 'key' feature of the i'th patch and f_i is the output feature of the i'th patch. We follow Wang *et al*. [33] and apply a threshold operation to the elements of matrix W. Specifically, we set any $W_{ij} \geq \tau^{\text{ncut}}$ to 1, and otherwise to $1e^{-5}$.

$$W_{ij} = \begin{cases} \frac{K_i K_j}{\|K_i\|_2 \|K_j\|_2} & \text{DINO model is used} \\ \frac{f_i f_j}{\|f_i\|_2 \|f_j\|_2} & \text{DINOv2 model is used} \end{cases} \tag{2}$$

Subsequently, we generate mask proposals by applying 1D K-means clustering [1] to the eigenvectors using every k value, ranging from 2 to k_{max}. This process resulted in the creation of n instance mask proposals per image, achieved by applying connected-components analysis to each segment produced by the K-means clustering.

Following this, we utilized an intersection over union (IoU)-based strategy to group masks into clusters. Given n proposal masks for a specific image, our clustering procedure begins with a greedy selection process. For each iteration, we identify the instance mask with the highest number of overlaps with other masks, surpassing an IoU threshold of $\tau^c = 0.6$. This mask is designated as the cluster pivot. The masks that share substantial IoU overlap with the pivot mask are considered part of the same cluster. Once a new cluster is formed, we systematically remove all associated masks and repeat this clustering procedure iteratively. This recursive process persists until all instance masks have been effectively associated with their respective clusters.

Formally, we denote the collection of clusters for the i-th image in the image set as $\mathcal{C}^i = \{C_1^i, C_2^i, \ldots, C_m^i\}$. In the context of each cluster C_j^i, we designate the mask members as $\{M_1^i, M_2^i, \ldots, M_p^i\}$. The resulting final mask for the cluster is determined as follows:

$$\text{Final Mask} = \begin{cases} 1, & \text{if } \frac{1}{p}\sum_{k=1}^{p} M_k^i > \tau^m \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Here, τ^m represents a threshold, and the final mask is set to 1 if the average of all masks for a given pixel within the cluster exceeds this threshold. τ^m sets the required consensus among the majority of masks to result a value of 1 in the final mask. We leverage a Conditional Random Field (CRF) [23] to perform post-processing on the final masks, facilitating the computation of their associated bounding boxes.

Once the clustering procedure is over, we compute the mask proposal score. This score, ranging from 0 to 1, is provided to each cluster j in each image i in the image set and is denoted by $y_{i,j}$. It corresponds to the cluster yielding the highest consensus mask value among all mask proposals of the same cluster size:

$$y_{i,j} = \frac{\left|C_j^i\right|}{\max\left(\left|C_1^i\right|, \left|C_2^i\right|, \ldots, \left|C_m^i\right|\right)} \tag{4}$$

As demonstrated in the following section, we showcase the applicability of this score in training a model using our innovative loss function applied at the instance level. This approach enables us to utilize all suggested masks without concern for inaccuracies, as those with lower scores will have a minor impact on the model's performance.

Based on this procedure, many clusters of VoteCut receive a score close to zero and, as such, have minimal contribution to the loss function. To minimize the calculation time, in cases where there are more than 10 VoteCut clusters, we remove the masks with the lowest scores.

3.3. Soft loss function

Given the aforementioned $y_{i,j}$, the score for the j-th pseudo-labeled instance in the i-th image, the corresponding bounding box loss is formulated as follows:

$$L_{box} = \sum_{i \in I} \sum_{j \in G_i} y_{i,j} L_{box,orig}^j \tag{5}$$

Here, I denotes the image set, G_i denotes the set of instances (i.e., masks) that are associated with the i-th image, and $L_{box,orig}^j$ is the original loss for the j-th box of the instance using input image x_i. Similarly, the mask loss is defined as:

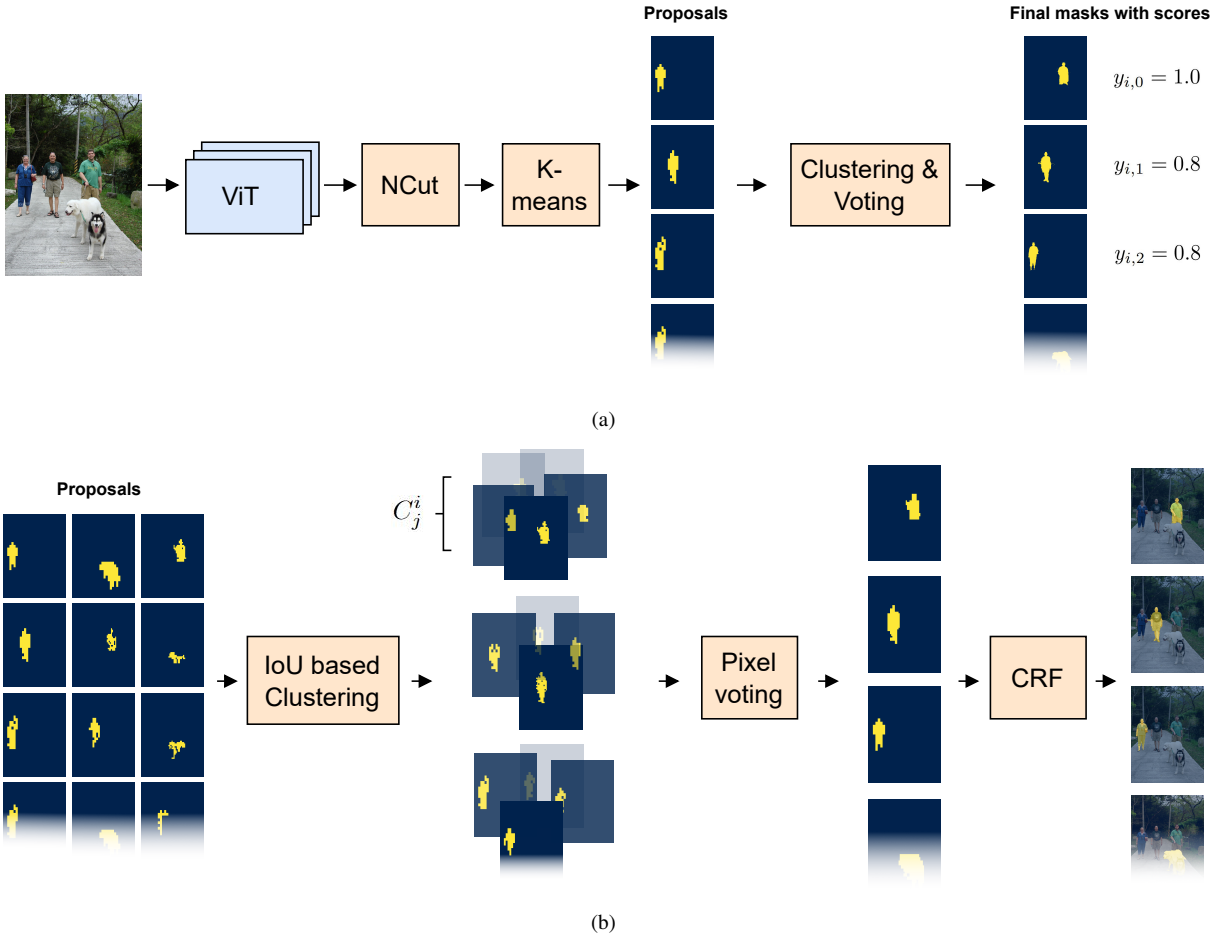

Figure 1. **(a)** An illustrated overview of the VoteCut workflow. A set of models initially makes inferences on the input image, producing feature representations for individual patches. Subsequently, Normalized Cuts (NCut) are performed following the methodology in [34], yielding the second smallest eigenvectors from each model. Multiple segment proposals are generated by applying 1D K-means clustering to these eigenvectors with varying K values. The final stage of VoteCut involves clustering these proposals and extracting definitive masks from each cluster via voting. Each definitive mask is also associated with a score. **(b)** The "Clustering & Voting" stage of VoteCut is detailed. First, segments are clustered using an Intersection over Union (IoU) threshold, determining segment membership within clusters. A voting mechanism is employed within each cluster to decide whether each patch should be included in the segment. Lastly, a Conditional Random Field (CRF) [23] is applied to refine the mask at a finer level. The cluster size determines the score assigned to each mask, as elucidated in Eq. (4).

$$L_{mask} = \sum_{i \in I} \sum_{j \in G_i} y_{i,j} L^j_{mask,orig} \tag{6}$$

For foreground score, a soft binary cross-entropy is employed:

$$L_{cls} = \sum_{i \in I} \sum_{j \in G_i} y_{i,j} \log(\sigma_f(x_i)) + (1 - y_{i,j}) \log(\sigma_b(x_i)) \tag{7}$$

Where $\sigma_f(x_i)$ represents the softmax output for foreground and $\sigma_b(x_i)$ for background.

Lastly, inspired by Wang et al. DropLoss [33], we define $r_{i,j}$ as the predicted region with maximum overlap of τ^{IoU} against 'ground truth' instances. This leads to the comprehensive loss function:

$$L = \sum_{i \in I} \sum_{j \in G_i} \mathbb{1}(\text{IoU}^{max}_{r_{i,j}} > \tau^{\text{IoU}})(L_{cls} + L_{box} + L_{mask}) \tag{8}$$

Where $\text{IoU}^{max}_{r_{i,j}}$ signifies the highest IoU with all pseudo-labeled instances. This loss avoids penalizing the model for missing 'ground-truth' objects, fostering exploration of diverse image regions. Similarly to Wang et al.'s work [33],

we applied a low threshold of $\tau^{\text{IoU}} = 0.01$.

3.4. CuVLER

Following the initial training stage, we conduct class-agnostic detection following the methodology outlined in [29]. This involves training a detector in a class-agnostic manner (CAD), utilizing the masks and the scores generated by VoteCut. It is essential to notice that CuVLER is trained solely on the ImageNet validation dataset. Thus, different datasets, such as the COCO dataset, can be considered candidates for zero-shot performance evaluation.

Self-training on a different domain

To generate pseudo-labels for a new dataset, we produce them using the CuVLER model inference. Then, we filtered out instances with confidence scores lower than 0.2. This newly curated dataset is utilized for further training, incorporating the updated confidence scores within our soft loss framework, resulting in the refinement of mask predictions.

3.5. Implementation details

Pre-processing stage When using DINO models, we resized the images to 480x480 pixels, and when using DINOv2 models, we resized the images to 476x476 pixels.

VoteCut The complete list of the utilized models appears in the supplementary materials. Unless specified otherwise, we utilized all aforementioned models and set $\tau^m = 0.2$ and $k_{max} = 3$. Further ablations related to these hyperparameters can be found in Sec. 5. We set $\tau^{cut} = 0.15$, as done by Wang *et al*. [33].

We employ a two-step resizing approach to ensure accurate pixel alignment between proposals. First, we resize all proposals to a fixed size before the IOU clustering phase. Then, after the pixel voting phase, we resize the final mask to match the image shape.

Training details All experiments were performed using the Detectron2 [35] platform, using a batch size of 16 and the copy-paste augmentation [15, 17]. Cascade Mask R-CNN [6] detector is used for CAD.

4. Experiments

This section delves into our experimental framework and is designed to evaluate our method's performance comprehensively. We divide the experiments into three essential evaluations, each presenting a different side of our methodology. First, we scrutinize the effectiveness of our method 'in domain' - within the domain for which the ViT models have initially trained. This provides us with a performance assessment in a familiar context. Second, we venture 'out of domain' zero-shot evaluation to assess the model's generalization capabilities (i.e., examining its adaptability to new environments). Lastly, we examine the dynamics of 'self-training' within an alternative domain. This evaluates the efficacy of unlabeled images from the domain of interest to enhance object discovery tailored to that domain. These experiments collectively offer a thorough understanding of the strengths and limitations of our approach in diverse settings, providing valuable insights for its practical applications.

The assessment of unsupervised object detectors presents a unique set of challenges. Primarily, these models lack an inherent understanding of semantic classes, rendering them unsuitable for evaluation through class-aware detection metrics. Consequently, we adopt the class-agnostic detection evaluation paradigm in line with prior research [4, 29, 33, 34]. Secondly, object detection datasets typically provide annotations for only a subset of the objects present in the images. Similarly to the work of Wang et al. [33], we have incorporated the Average Recall (AR) metric to address this limitation. AR proves valuable in assessing unsupervised detection models as it refrains from penalizing them for detecting novel, unlabeled objects within the dataset.

4.1. In-domain evaluation

In this experiment, we conducted an 'in-domain' evaluation of the ImageNet validation set. We chose this specific dataset to align with the domain on which the ViT model was originally trained. Our comparative analysis unfolds through two distinct scenarios: (1) 'no CAD', where we solely generate masks using the features extracted from the pre-trained model, abstaining from training a dedicated detector; (2) 'with CAD', where we train a detector using the masks we create and subsequently deploy it for class-agnostic object detection. Keeping with established conventions, our evaluation employs the widely recognized COCO metrics. However, given that only a fraction (approximately 10 %) of ImageNet has bounding-box annotations and none have segmentations, we exclusively report the performance of bounding-box metrics.

Our approach introduces a novel scoring mechanism, detailed in Eq. (4), which allows for a more detailed and insightful assessment of our model's performance. To facilitate a fair and equitable comparison, we incorporated DINOv2 models within our methodology. Consequently, we compared the previous state-of-the-art (SOTA) and our best-performing DINOv2 model. For reference, the 'no-CAD' method of the previous SOTA was set to 1.0 since this methodology does not provide a direct score evaluation.

It's worth noting that this additional comparison does not yield any significant improvement in favor of the previous SOTA; this is shown in Table 1, where MaskCut† utilizes the best-performing DINOv2 model, instead of the DINO model used in the original MaskCut [33].

As depicted in Table 1, in the 'no CAD' scenario, Vote-

Method	CAD	AP	AP_{50}	AP_{75}	AR_{100}
TokenCut[34]	×	14.4	27.0	13.4	26.9
MaskCut[33]	×	10.6	20.3	10.0	27.7
MaskCut†[33]	×	8.3	14.9	7.7	22.8
VoteCut (ours)	×	**20.9**	**36.2**	**20.0**	**45.0**
CutLER[33]	✓	29.2	48.8	29.8	56.7
CuVLER (ours)	✓	**33.2**	**52.6**	**33.7**	**59.0**

Table 1. In-domain evaluation on the ImageNet validation set. The comparative analysis is divided into two scenarios: 'no CAD' and 'with CAD'. Keeping with established conventions, the evaluation employs the COCO metrics. We exclusively report the performance of bounding-box metrics. †: utilizing our best-performing DINOv2, instead of the DINO model used in MaskCut [33].

Cut (our approach) showcases substantial improvements, with performance enhancements ranging from approximately 60% to 100% across various metrics. In the 'with CAD' scenario, we observed more modest yet significant improvements, ranging from 4% to 13% across all metrics. These results underscore our approach's competence and efficiency in 'in-domain' evaluations.

Figure 2 illustrates our proposed methodology performance compared to other SOTA models.

4.2. Zero-shot evaluation

In this experiment, we assessed our methodology performance across seven diverse benchmarks, detailed in the Supplementary Materials. Following the methodology of the previous SOTA [33], we employ a cascade Mask R-CNN model trained exclusively on ImageNet, a methodology referred to as 'zero-shot' due to its singular domain training and cross-domain evaluation without further adaptation. Our evaluation is based on the COCO metrics, encompassing Average Precision (AP) and AP_{50} scores. Detailed results for all benchmarks are available in the Supplementary Materials.

As depicted in Table 2, our approach demonstrates significant improvements of up to 20%. We consistently observe performance enhancements across all benchmarks, except for the Cliparat dataset, where a minor decline of 1% is noted in the AP_{50} metric. Importantly, our method achieves superior performance over the previous SOTA after a single training epoch, obviating the necessity for extensive self-training stages on the ImageNet dataset.

4.3. Self-training evaluation

In this experiment, we harness our self-training methodology, which leverages unlabeled images from the domain of interest and subjected it to a rigorous comparison with previous approaches. Aligning with established practices, we train a Cascade Mask-RCNN model on the COCO train2017 dataset and evaluate our results on widely recognized benchmarks for bounding-box and segmentation tasks.

In Table 3, we present a comparison of our detector's performance on two prominent benchmarks: COCO `val2017` and COCO `20K`, the latter being a subset of 20,000 images from the COCO dataset [25, 29, 32, 33]. Notably, our results showcase improvements in all metrics, with enhancements reaching up to approximately 10%.

We have expanded our evaluation to a more challenging benchmark - LVIS [19], which encompasses over 1,000 entry-level object categories and naturally exhibits a long-tailed data distribution. We found that the improvement was more modest, with enhancements of up to 7% (see Table 4). This outcome aligns with our expectations, considering the significant shift in category distribution between LVIS and ImageNet. Additional insights and detailed explanations regarding these benchmarks are available in the Supplementary Materials.

5. Ablations

In Table 5, we present an ablation study on the COCO `val2017` dataset [25], aiming to illustrate the importance of each component. Utilizing VoteCut with CAD training, as an alternative to the baseline MaskCut [33], after CAD training, results in a 12% and an 18% improvement in AP_{50}^{mask} and AP^{mask} respectively. Additionally, utilizing the soft target loss further enhances the results, with a notable 9% increase in AP_{50}^{mask} and 7% in AP^{mask}. We emphasize that the results achieved up to this point were obtained without reliance on the COCO dataset, thus establishing a zero-shot evaluation setup. Furthermore, incorporating data from the COCO dataset using the self-training stage, as detailed in Sec. 3.4, yields an additional 5% boost for AP_{50}^{mask} and a 6% improvement in AP^{mask}. Collectively, these components produce a noteworthy 29% increase in AP_{50}^{mask} and a substantial 35% enhancement in AP^{mask}, highlighting their combined impact on segmentation quality.

We evaluate the VoteCut method (without CAD) in the following ablation studies on the ImageNet validation set. [25] dataset, in an in-domain setup. These studies aim to investigate the effects of individual hyperparameters while holding the remaining parameters constant.

τ^m introduced in Eq. (3) acts as an integral threshold within the proposed method. In Figure 3, we demonstrate the impact of varying threshold values. Configuring τ^m at 0.2 is strictly dominant across all evaluated metrics.

k_{max} is an integral part of the VoteCut method, directly impacting the number of generated proposals. In Figure 4, we aim to illustrate the influence of varying k_{max} values. Notably, $k_{max} = 3$ excels in AP-based metrics and ranks almost equally well in AP^{box}, with under a one percent difference from the top result.

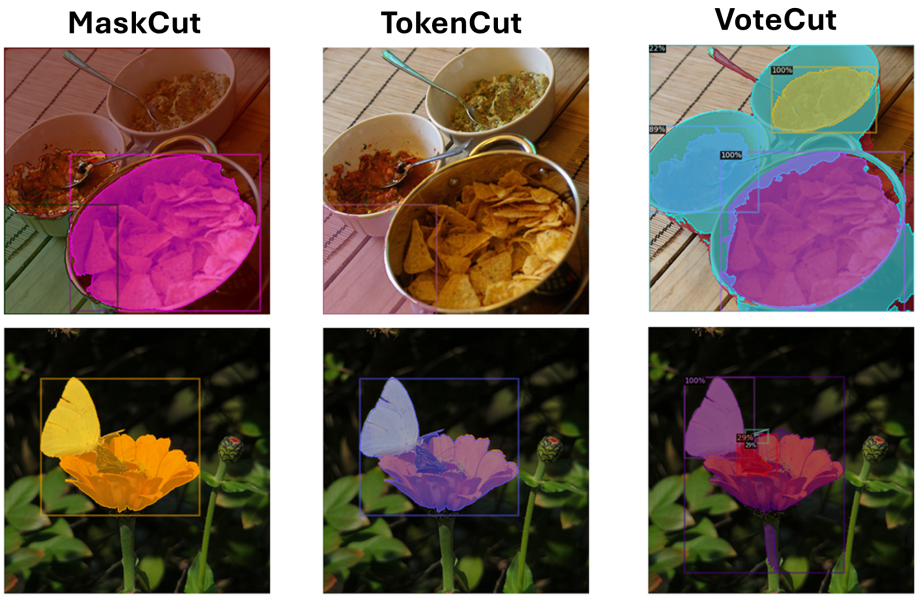

Figure 2. Visual illustration of VoteCut performance vs. SOTA NCut based object-discovery methods on the ImageNet validation set. The VoteCut bounding box score is calculated according to Eq. (4)

Method	COCO		COCO20K		VOC		OpenImages		Clipart		Watercolor		Comic	
	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP	AP_{50}	AP
Prev. SOTA [33]	21.9	12.3	22.4	12.5	36.9	20.2	17.3	9.7	21.1	8.7	37.5	15.7	30.4	12.2
CuVLER (ours)	23.0	12.6	23.5	12.7	39.4	22.3	19.6	11.6	20.8	9.3	41.3	19.0	32.2	14.6
vs. prev. SOTA	**+1.1**	**+0.3**	**+1.1**	**+0.2**	**+2.5**	**+2.1**	**+2.3**	**+1.9**	-0.3	**+0.6**	**+3.8**	**+3.3**	**+1.8**	**+2.4**

Table 2. SOTA zero-shot unsupervised object detection performance on seven datasets. The reported results are based on the COCO metrics, encompassing both Average Precision (AP) and AP_{50} scores. The presented models are trained in an unsupervised manner solely on ImageNet. Results of [33] are produced with official code and checkpoint.

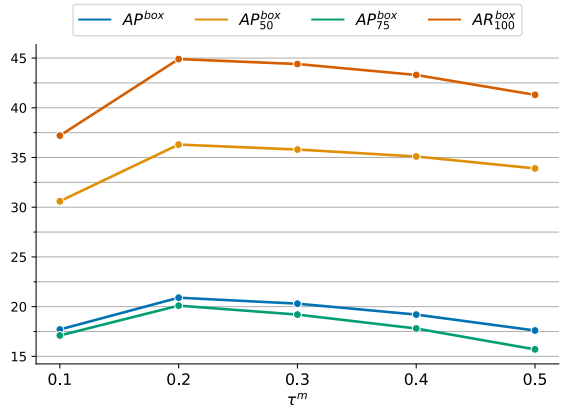

Figure 3. In-domain evaluation of the VoteCut method, without CAD training, with varying τ^m on the ImageNet validation set.

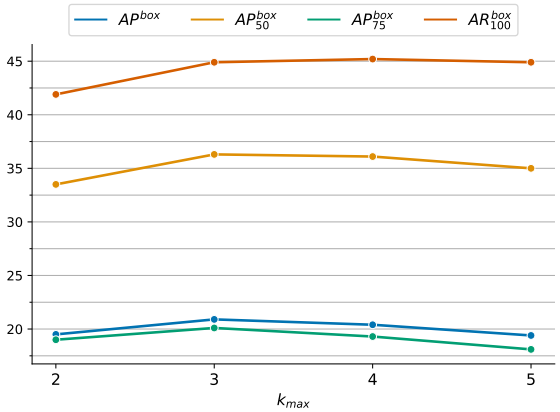

Figure 4. Results of the VoteCut method without CAD training in an in-domain configuration with different k_{max} values on the ImageNet validation set.

We observe from the provided ablation studies that the proposed method demonstrates robustness to variations in both τ^m and k_{max}, as even suboptimal settings for these

Method	Detector	Init.	COCO 20K						COCO val2017					
			AP^{box}_{50}	AP^{box}_{75}	AP^{box}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}
LOST[29]	FRCNN	DINO	-	-	-	2.4	1.0	1.1	-	-	-	-	-	-
MaskDistill [31]	MRCNN	MoCo	-	-	-	6.8	2.1	2.9	-	-	-	-	-	-
FreeSOLO [32]	SOLOv2	DenseCL	9.7	3.2	4.1	9.7	3.4	4.3	9.6	3.1	4.2	9.4	3.3	4.3
CutLER [33]	Cascade	DINO	22.4	11.9	12.5	19.6	9.2	10.0	21.9	11.8	12.3	18.9	9.2	9.7
CuVLER† (ours)	Cascade	DINO	**24.1**	**12.3**	**13.1**	**21.6**	**9.7**	**10.7**	**23.5**	**12.0**	**12.8**	**20.4**	**9.6**	**10.4**

Table 3. Unsupervised object detection and instance segmentation on COCO `20K` and COCO `val2017`. We report the detection and segmentation metrics and note the detectors (Detector) and backbone initialization (Init.). All models results are obtained with the official code and checkpoint. †: model was further self-trained on the target domain.

Method	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
CutLER[33]	4.5	8.4	3.9	3.5	6.7	3.2
CuVLER†	**4.7**	**8.9**	**4.1**	**3.8**	**7.2**	**3.4**

Table 4. Evaluation on the LVIS benchmark.†: model was further self-trained on the target domain.

Methods	AP^{mask}_{50}	AP^{mask}
MaskCut CAD [33]	15.8	7.7
+VoteCut CAD	17.7	9.1
+Soft target loss (CuVLER)	19.3	9.8
+Self-training stage	20.4	10.4

Table 5. Component ablation study of our methodology. We illustrate the impact of each component on the COCO `val2017` dataset.

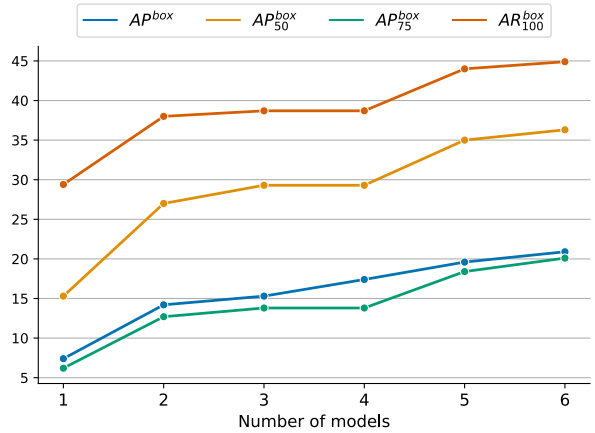

Figure 5. Model count ablation test. The results are obtained in an in-domain setup on the ImageNet validation set using the VoteCut method without CAD training.

parameters outperform TokenCut [34] by a significant margin.

In Figure 5, we can discern the impact of employing an increased number of models. The results correspond to maximum obtained by calculating all possible combinations of models within our set, with each calculation constrained by the number of selected models. A clear trend emerges, indicating that employing a greater number of models leads to improved outcomes. This trend suggests that each model captures distinct information, and the collective strength of the models surpasses the performance of each model in isolation.

6. Conclusion and Limitation

We have presented VoteCut, a novel method for unsupervised object discovery capable of generating a dynamic number of mask instances with corresponding confidence scores, outperforming previous counterparts. We also introduced CutLER, a zero-shot model that enhances VoteCut results using a self-training phase with a novel soft loss. Furthermore, we presented an additional enchantment based on self-training that improves cross-domain results. CutLER outperforms the previous SOTA in both zero-shot and unsupervised setups across multiple datasets.

While our hyperparameter studies highlighted the robustness of our method — with optimal configurations for parameters like τ^m and k_{max} substantially influencing results — we also observed the inherent strength of ensemble techniques. For instance, as the number of models increased, there was a clear trend of performance improvement, signaling the importance of diversified model input.

Limitation. The computational requirements of integrating multiple models within the VoteCut framework may present challenges in resource-constrained settings. However, leveraging VoteCut for training a single segmentation model, as showcased by CuVLER, mitigates this constraint while enhancing segmentation performance. Additionally, utilizing the ImageNet dataset for pseudo-label generation may introduce biases, given its simplified domain and lower object density, potentially reducing false positives. Future research should investigate how source domain characteristics impact pseudo-label quality and inference outcomes.

References

[1] David Arthur and Sergei Vassilvitskii. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007. 3

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 2

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[4] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14605–14615, 2022. 5

[5] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. 2

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 5

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 2

[8] 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 IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1, 2

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021. 2

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[12] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004. 2

[13] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[15] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[17] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2918–2928, 2021. 5

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 2

[19] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 6

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[22] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018.

[23] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems*, 24, 2011. 3, 4

[24] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020.

[25] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[26] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020. 2

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3

[28] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 1, 2, 3

[29] 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. 2021. 2, 5, 6, 8

[30] Mariano Tepper, Pablo Musé, Andrés Almansa, and Marta Mejail. Automatically finding clusters in normalized cuts. *Pattern Recognition*, 44(7):1372–1386, 2011. 2

[31] Wouter Van Gansbeke, Simon Vandenhende, and Luc Van Gool. Discovering object masks with transformers for unsupervised semantic segmentation. *arXiv preprint arXiv:2206.06363*, 2022. 2, 8

[32] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 2, 6, 8

[33] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[34] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv preprint arXiv:2209.00383*, 2022. 2, 3, 4, 5, 6, 8

[35] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[36] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 2

[37] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016. 2

[38] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30 (11):3212–3232, 2019. 1

[39] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023. 1