Unsupervised Semantic Object Localization

Jay Khinchi, Subhashi Jayant, Mehar Khatoon, Risheek Lahiri, Bhushan Patil
Group: D-Learner
Deep Learning (CS590)
IIT Guwahati

***Abstract* — Recent advances in self-supervised visual representation learning have paved the way for unsupervised methods tackling tasks such as objects. We are proposing a heuristic approach of self-supervision for instance segmentation without any supervision. First, we generate multiple binary masks per image using self-supervised features from DINO (Token-cut Algorithm). Second, we implement a dynamic loss-dropping strategy which can learn a detector from the initial masks while encouraging the model to explore objects missed initially. Third, we further improve the performance of our method through multiple rounds of self-training.**

I. Introduction

Object Localization is the task of locating an instance of a particular object category in an image, typically by specifying a tightly cropped bounding box centered on the instance. Our model was trained exclusively on unlabeled ImageNet data without needing additional training data. First, the data is passed to ViT which learns self-supervised features. Then, using these pretrained self-supervised features we produce multiple initial coarse masks for each image. Second, we apply a dynamic loss dropping strategy that can learn a detector from initial masks while encouraging the model to explore objects missed initially. The detectors clean the initial coarse masks and produce masks (and boxes) that are better than the coarse masks used to train them. Multiple rounds of self-training on the models' own predictions allow it to evolve from capturing the similarity of local pixels to capturing the global geometry of the object, thus producing finer segmentation masks.

II. Related Work

Unsupervised object localization, traditionally, follows two paths. First, unsupervised saliency detection methods where we find binary masks of objects. And second, unsupervised object detection where we try to find bounding boxes around objects.

FOUND [1] uses self supervised features to partition background and foreground patches. A pixel that does not belong to the background is likely to belong to an object. It does not make hypotheses about the number or the size of objects in order to find them.

In the context of "Localizing Objects with Self-Supervised Transformers and no Labels," LOST (Local Object Self-Tracker) [2] is the primary method proposed for localizing objects within images without the use of labeled data. LOST is a self-supervised approach that relies on features extracted from a pre-trained visual transformer, specifically a ResNet50 model pre-trained with DINO (Data-Efficient Image Transformer) self-supervision. Here's an overview of how the LOST method works:

> Feature Extraction from Pre-trained Vision Transformer

> Patch Correlations for Object Localization
> Box Extraction
> Unsupervised Object Detection
It faces limitations when dealing with overlapping instances of objects and when an object covers a significant portion of the image.

K-means for unsupervised instance segmentation using a self-supervised transformer [3] divides the image into two subgraphs (foreground and background) using the heuristic method or graph cut algorithm then the k-means is applied to divide the feature of patches belonging to foreground and background. A vision transformer considers a given image of size as input by splitting it into non- $H \times W$ overlapping size patches. Then the extracted feature of the dataset is trained by bisecting k-means which generates a noisy object mask from a single image feature. These courses are refined by bisecting k-means. But this method does not work in Euclidean distance and detection of the main object is challenging.

DINO [4] observes that the underlying semantic segmentation of images can emerge from the self-supervised ViT, which does not appear explicitly in either supervised ViT or ConvNets. Based on this observation, LOST [2] and TokenCut [5] leverage self supervised ViT features and propose to segment one single salient object [2,5] from each image based on a graph that is constructed with DINO's patch features. However, DINO and TokenCut can not detect more than one object from each image; TokenCut and LOST can not improve the quality of features for better transfer to downstream detection and segmentation tasks. Unlike these works, our approach can locate multiple objects and serve as a pre-trained model for label efficient and fully-supervised learning.

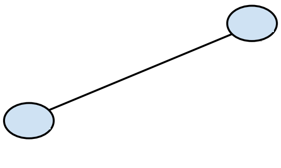
III. Proposed Work and Implementation

Detector Model

- Used ResNet-50 architecture.
- Generate coarse mask using MaskCut approach.
- Learn a detector on this mask using a robust Loss function.

First, we propose MaskCut which generates multiple binary masks per image using self-supervised features from DINO. Second, we will show a dynamic loss-dropping strategy, called DropLoss, which can learn a detector from MaskCut's initial masks while encouraging the model to explore objects missed by MaskCut. Third, we further improve the performance of our method through multiple rounds of self-training.

Normalized cut : Construct graph i.e take image as node.



Wij =measure similarity of the connected node.

(D-W)x = λDx; D=(n*n)diagonal matrix; x= eigen vector; $d(i) = \sum j\,Wij$; W = n*n symmetric matrix. $Wij=(ki*kj)/(|ki|_2|kj|_2)$ Ki and kj are key features of patch i and j respectively. Solve the above equation and find the second smallest eigenvector x.

$Mij = \{1, \text{if } Mij^t >= mean(x^t)\ 0, \text{otherwise}\}$

After getting bipartition x^t from Ncut at stage t we get two disjoint groups. To determine which group corresponds to the foreground, we make use of two criteria:

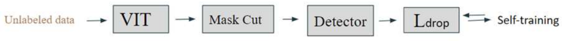
Intuitively, the foreground patches should be more prominent than background patches. Therefore, the foreground mask should contain the patch corresponding to the maximum absolute value in the second smallest eigenvector M^t. We incorporate a simple but empirically effective object-centric prior, the foreground set should contain less than two of the four corners.

Binary mask M^t where: Foreground mask should contain $\max\{abs(M_2)\}$ [second smallest]. If above not satisfied then $M^tij = 1-M^tij$ (Reverse partitioning of foreground and background). If both above not satisfied then set $Wij < \boldsymbol{\lambda}^{ncut}$ to $1*e^{-5}$; $Wij >= \boldsymbol{\lambda}^{ncut}$ to 1. To get mask for $(t+1)^{th}$ object, update node $W^{t+1}ij$ via masking out these nodes corresponding to the foreground in the previous state. $W^{t+1}ij = (ki\boldsymbol{\Pi}^t_{s=1}Mij^s)(kj\boldsymbol{\Pi}^t_{s=1}Mij^s) / |ki|_2|kj|_2$.

Droploss for Exploring Image regions: We ignore the predicted region r_i that have a small overlap with the ground truth. $L_{drop}(ri)=1(IoU_i^{max}>\tau^{IoU})L_{vanilla}(ri)$. We used $\tau^{IoU} = 0.01$

Multi Round self Training: Detector refine mask quality and our droploss strategy then to discover new object masks. Use multiple rounds of self training to improve the detector performance. In our approach three rounds of self training is sufficient to obtain good performance.

IV. EXPERIMENTAL DETAILS



We use MaskCut with three stages on images resized to 480×480 pixels and compute a patch-wise affinity matrix using the ViT-B/8 and DINO model. We Used transfer learning self-supervised transformer for unsupervised object discovery using normalized cut (Token Cut Algorithm).

Trained the detector with a ResNet-50 backbone. Initialize the model with the weights of a self-supervised pretrained DINO model. Randomly downsample the mask with a scalar uniformly sampled between 0.3 and 1.0. Optimize the detector for 160K iterations using SGD with a learning rate of 0.005, which decreases by 5 after 80K iterations, and a batch size of 16.

Initialize the detection model in each stage using the weights from the previous stage. Optimize the detector using SGD with a learning rate of 0.01 for 80K iterations. The self-training stage can provide a sufficient number of pseudo-masks for model training; we didn't use the DropLoss during the self-training stages.

As a baseline, we follow the settings from MoCo-v2 and train the same detection architecture initialized with a MoCo-v2

ResNet50 model, given its strong performance on object detection tasks. MoCo-v2 and our model uses the same training pipeline and hyper-parameters and are trained for the 1× schedule using Detectron2 [6], except for extremely low-shot settings with 1% or 2% labels. Our detector weights are initialized with ImageNet-1K, except for the weights of the final bounding box prediction layer and the last layer of the mask prediction head, which are randomly initialized with values taken from a normal distribution.

We have used the COCO 20k dataset. It is a large-scale object detection and instance segmentation dataset, containing about 115K and 5K images in the training and validation split, respectively.

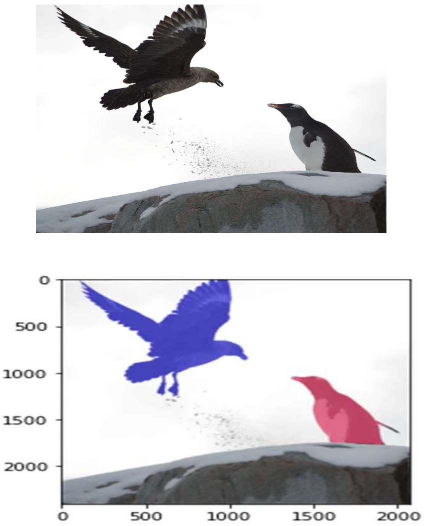


Fig1. Input image (above) to output image (below) of Maskcut (Tokencut algo).

V. RESULTS

Methods	Pretrain	Detector	Initialization	COCO 20K					
				APbox50	APbox75	APbox	APmask50	APmask75	APmask
FreeSOLO	IN+COCO	SOLOv2	DenseCL	9.7	3.2	4.1	9.7	3.4	4.3
DINO	IN	-	DINO	1.7	0.1	0.3	-	-	-
TokenCut	IN	-	DINO	-	-	-	-	-	-
Our Method	IN	Cascade	DINO	22.4	12.5	11.9	19.6	10.0	9.2
Vs. prev. SOTA				+12.7	+9.3	+7.8	+9.9	+6.6	+4.9

Table 1. Unsupervised object detection and instance segmentation on COCO 20K

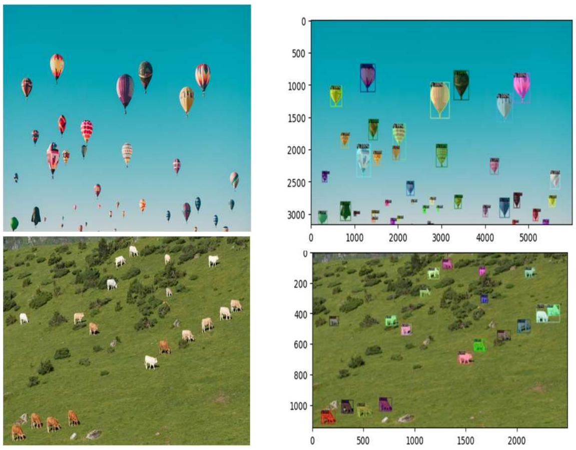


Fig2. Input images we gave to our model (left) and the output generated (right)

VI. ABLATION STUDY

We used cascade R-CNN detection architecture and evaluated our model on the dataset mentioned above. We trained the model on ImageNet. Using TokenCut it only segments a single instance per image. In order to generate more than one segmentation mask per image, we modified it by using more of the smaller eigen vectors and combining all produced masks. Our model works with various architectures and its performance is improved with stronger architecture.We trained DINO and our method on the same dataset.

Methods	APbox 50	APbox	ARbox 100	APmask 50	APmask	ARmask 100
TokenCut (1 ev.)	5.2	2.6	5.0	4.9	2.0	4.4
TokenCut (3 ev.)	4.7	1.7	8.1	3.6	1.2	6.9
MaskCut (t = 3)	6.0	2.9	8.1	4.9	2.2	6.9
Our Method	21.9	12.3	32.7	18.9	9.7	27.1

Our model achieves much higher results even when compared to a modified TokenCut that can produce more than one mask per image. Compared to TokenCut, MaskCut gets a higher recall without reducing precision. We report results on COCO.

Size →	240	360	480	640
APmask 50	15.1	16.6	17.7	17.9

(a) Image size.

(a) We vary the size of the image used for MaskCut.

τ ncut →	0	0.1	0.15	0.2	0.3
APmask50	17.1	17.5	17.7	17.6	17.5

(b) τ ncut for MaskCut.

(b) We vary the threshold τ ncut in MaskCut, which controls the sparsity of the affinity matrix used for Normalized Cuts.

N →	2	3	4
APmask 50	16.9	17.7	17.7

(c) # masks per image.

(c) We vary the number of masks extracted using MaskCut.

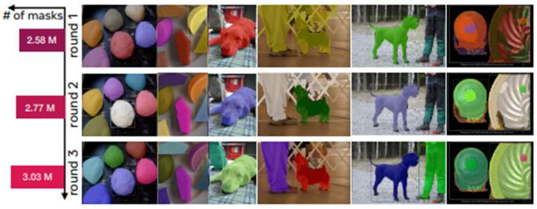
τ IoU →	0	0.01	0.1	0.2
APmask 50	17.4	17.7	14.4	12.7

(d) τ IoU for DropLoss

(d) We vary τ IoU in DropLoss, i.e., the maximum overlap between the predicted regions and the ground truth beyond which the loss for the predicted regions is ignored.

	UVO			COCO		
	APmask 50	APmask	APmask 75	APmask 50	APmask	APmask
1 round	20.6	9.0	7.0	17.7	8.8	8.0
2 rounds	22.2	9.6	7.5	18.5	9.5	8.8
3 rounds	22.8	10.1	8.0	18.9	9.7	9.2
4 rounds	22.8	10.2	8.2	18.9	9.8	9.3

Number of self-training rounds



Multiple rounds of self-training can improve the pseudo masks in terms of quality and quantity.

	Mask R-CNN	Cascade Mask R-CNN	ViTDet
APbox 50 / APbox	20.3 / 10.6	20.8 / 11.5	21.5 / 11.8
APmask 50 / APmask	17.2 / 8.5	17.7 / 8.8	18.0 / 9.0

Our model with different detection architectures.

VII. REFERENCES

[1] Oriane Siméoni, Chloé Sekkat, Gilles Puy Antonín Vobecký, Éloi Zablocki, Patrick Pérez. Unsupervised Object Localization: Observing the Background To Discover Objects. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 2023, pp. 3176-3186.

[2] Oriane Sim\'eoni and Gilles Puy and Huy V. Vo and Simon Roburin and Spyros Gidaris and Andrei Bursuc and Patrick P\'erez and Renaud Marlet and Jean Ponce. Localizing Objects with Self-Supervised Transformers and no Labels. *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

[3] Lim, SeungTaek and Park, JaeEon and Lee, MinYoung and Lee, HongChul, K-Means for Unsupervised Instance Segmentation Using a Self-Supervised Transformer. In *SSRN*, 2022. Available: https://ssrn.com/abstract=4251338

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve J ´egou, 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.

[5] 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.

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