Object Detection
Comparative Exploration of Two Different Worlds

Rishi Dey Chowdhury

Indian Statistical Institute, Kolkata

Introduction

Object detection is a fundamental problem in computer vision that has been extensively studied over the past two decades. However, recent advancements in deep learning techniques such as R-CNN [1], YOLO [2], and SSD [3] have renewed interest in this field. With availability of moderate sized object detection datasets like COCO [4], LVIS [5] and Flickr30k [6], computer vision researchers have been able to train SOTA object detection models. But there have always been a significant gap between supervised and unsupervised methods [7] to train such models. Here in this work, we present a comparison between object detection techniques from each of these directions and aim to examine the gap in their performance. Moving towards unsupervised techniques is often desirable due to zero requirement of human annotations for training such systems.

In particular we focus on two different techniques (a) Weakly supervised method of training as adopted in Detic [8] which relies on CenterNet2 [9] architecture and (b) Unsupervised method like Cut-and-Learn [10] which under the hood rely on Self-Supervised Vision Transformer [11] for generating the object annotations without human intervention and training a detector like Cascade R-CNN [12] on these unsupervisedly annotated dataset. The major difference between these two methods is primarily the nature of their training datasets, where the prior focuses on open-vocabulary object detection [13] for pushing the number of detected classes to over 21K and the latter targets the problem of object localization solely using vision data with no supervision.

We compare both the techniques based on their performance on the COCO Validation Split 2017 (Dataset Link) and we observe from Table 1, that the gap between unsupervised and supervised object detection has decreased significantly. Even more surprising is how the unsupervised features are more information rich and surpasses the performances of supervised object detection model consistently across all the metrics.

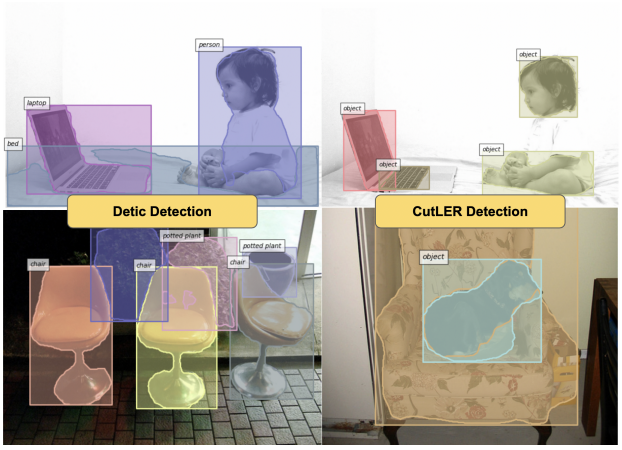

Figure 1: *Object Detection using Detic and CutLER. Detic performs both object recognition and object detection, whereas CutLER can do only object detection. Although CutLER can be fine-tuned with supervised approaches and the results are shown in Table 1*

Related Work

Weakly Supervised Object Detection This involves using image level labels for training object detectors without any bounding box supervision [13, 14, 15, 16]. Several existing techniques rely on generating region proposals [17, 18], clustering similar features and assigning labels to these clustered region proposals [19]. These weak labels can be treated as training data for the object detector. Since, no bounding box supervision is used these methods rely on low-level region proposal techniques [20, 21], thereby limiting the localization quality.

Another strategy adopted by several works is semi-supervised weak supervision [22, 23, 24, 25, 26, 27, 28]. In YOLO9000 [29], a mix of classification data and detection data is used in every mini-batch to assign classification labels to anchors with highest predicted score. This is very similar to the self-training [30] approach and the clustering-based approach [19] discussed above.

Self-Supervised Learning Self-Supervised Learning is often treated as a pretraining technique, where a model is trained to learn feature representation for various image patches obtained from the same image. This does not require any supervision and with approaches like Contrastive Learning [31, 32,

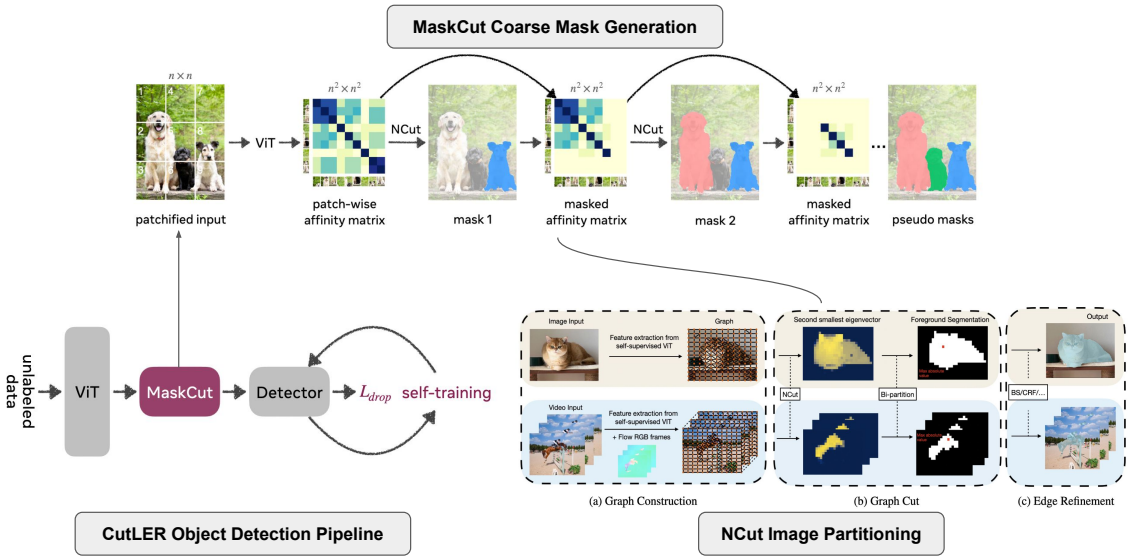

Figure 2: *Object Detection using CutLER [10]. Some of the figures are obtained from the Cut and Learn for Unsupervised Object Detection and Instance Segmentation paper*

Methods	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{box}_{m}	AR^{box}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{m}	AR^{mask}
CutLER† [10]	**44.66**	**62.50**	**48.49**	**48.19**	59.30	**38.53**	**59.60**	**41.71**	**41.43**	**51.40**
CutLER [10]	12.32	21.97	11.89	12.72	32.8	9.78	18.91	9.19	8.77	27.10
Detic [8]	43.50	60.85	47.19	47.44	**59.70**	35.79	57.00	38.31	40.16	50.50

Table 1: *Reproduction of evaluation results on COCO Validation Split 2017. CutLER† is the object detector fine-tuned on* 100% *labelled COCO training dataset with initialized unsupervised CutLER weights. Thus the training setting for CutLER† is similar to that of Detic viz trained on LVIS + COCO*

33, 34], Teacher-Student Knowledge Distillation [35, 36] and Masked Image Modelling [37, 38] are able to generate superior image features. Recently, the backbone of choice for such pretrainining has been Vision Transformers [11, 39] inspired by the success of its counterpart in the language domain [40]. The key benefits of using such pretraining is that the features for the image patches corresponding to the same object are usually mapped closer in the vision embedding space reflected through high cosine similariry [41, 42] and that of dissimilar objects are scored lower. This often results in better clustering of similar patch-level object features and attention heatmaps also shows evidence for such learning by revealing semantic maps and object boundaries without any supervision [35, 36].

Unsupervised Object Detection Pretrained self-supervised transformer-based models like DINO [35] and DINOv2 [36], result in explicit segmentation of salient objects [43, 44, 45] based on TokenCut [45], LOST [44] and MaskCut [10] which are able to come up with proposals for objects which are further used for training object localization and detector models. These methods rely on treating the patched image as a graph that is constructed with the patch-based features from the self-supervised models. CutLER [10] uses this idea to generate annotated object detection dataset from vision data alone. These attention heatmap based object masks and detection can be further improved with self-training or fine-tuning with Weakly supervised datasets. Such techniques do class-agnostic object detection and requires supervision in terms of class annotated datasets to output the object classes for the detected objects. Grounding DINO [46] is a work in that direction where the DINO model is combined with grounded pretraining of object classes from COCO, etc allowing class-specific object detection.

Detic Although we had a brief discussion on Detic, we will take a better look at the object detection technique used by Detic here. Detic uses a simple method to leverage image-level supervision to learn a object detector, including for classes without any bounding box annotation. It combines two types of datasets to incorporate a semi-supervised weak supervision technique for training the model, which includes: (a)

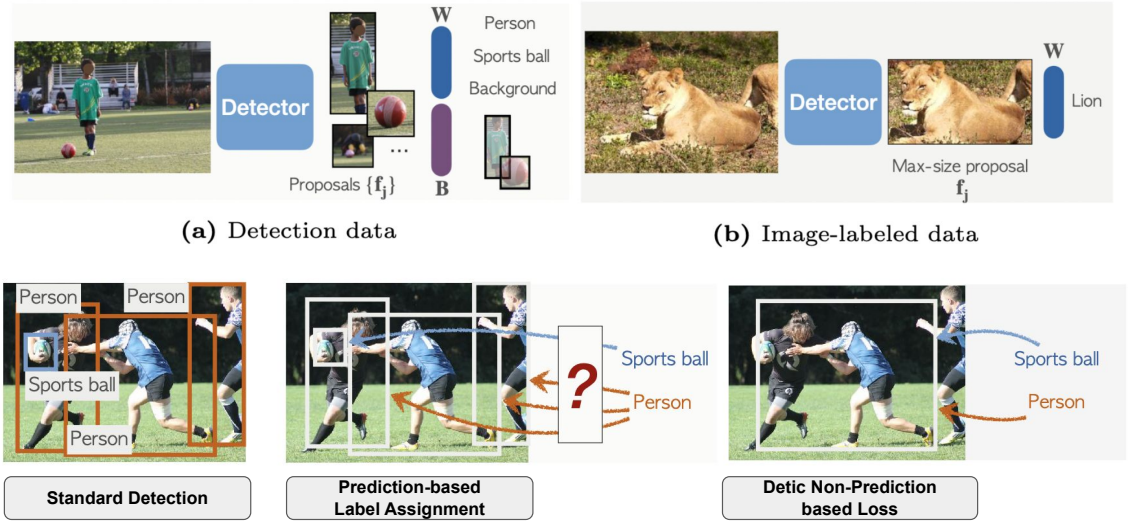

Figure 3: *The training dataset of Detic consists of batches of fully annotated images with bounding box + object classes and image level classifications respectively*

localization + recognition annotated data and (b) image level classification labels as shown in Figure 3. It also expands the performance of the detector to an open vocabulary setting by predicting labels closes to an output embedding for the recognized object class.

During training, each mini-batch is composed of images from both types of annotated subsets of data described above. For images with box labels, Detic is trained using standard two-stage detector training [47] and for image-level labeled images, Detic is trained with features from a fixed region proposal for image classification. Thus, Detic only computes the localization losses (RPN loss and bounding box regression loss) on images with ground truth box labels. Concisely, the loss function used by Detic during its training is as follows:

$$L(\mathbf{I}) = \begin{cases} L_{rpn} + L_{reg} + L_{cls} & \mathbf{I} \in \mathcal{D}^{det} \\ \lambda L_{max-size} & \mathbf{I} \in \mathcal{D}^{cls} \end{cases}$$

where, $\mathbf{I}$ is an image in the batch, $\mathcal{D}^{det}$ and $\mathcal{D}^{cls}$ are the sets corresponding to the images with bounding box + classification i.e. detection data and only classification data respectively. L_{cls} is standard cross-entropy classification loss, L_{reg} is the regression loss for the bounding box coordinates and the $L_{max-size}$ is the Cross-Entropy loss corresponding to the largest region proposed by the RPN.

For implementation, Detic uses a CenterNet2 Architecture and is trained on the Imagenet [48] and Google Conceptual Caption [49] datasets achieving SOTA result at its time and also pushing the total number of detected object classes to 21k placing it somewhere near the extreme classification setting.

Cut and Learn Cut and Learn is a completely unsupervised approach for object detection which starts with the pretrained patch-based feature embeddings of DINO and iteratively solves an eigenvalue system proposed by NCut [50, 51] (equation below), with the similarity feature matrix W being the attention weights and $D = \sum_j W_{ij}$, to partition the attention heatmap into two parts foreground and background.

$$(D - W)x = \lambda Dx$$

In each round of NCut, we mask the foreground by setting it to 0 and apply the same process to the background. This iterative NCut is termed as MaskCut step (Figure 2) in Cut and Learn. These masks form the coarse object masks and bounding boxes which is improved further by self-training. This simple straightforward technique achieves impressive results without any human annotations. The results are also summarized in Table 1

For association of class labels with these detected objects, the CutLER is fine-tuned on COCO and LVIS dataset leading to $CutLER^{\text{deg}}$. $CutLER^{\dagger}$ outperforms many supervised object detection model, thereby significantly closing the gap between supervised, weakly supervised and unsupervised object detection techniques.

The CutLER model is implemented using pretrained DINO for MaskCut step and Cascade RCNN [12] for the self-training and fine-tuning.

References

[1] Ross Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV].

[2] Joseph Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2016. arXiv: 1506.02640 [cs.CV].

[3] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. URL: https://doi.org/10.1007%2F978-3-319-46448-0_2.

[4] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[5] Agrim Gupta, Piotr Dollár, and Ross Girshick. *LVIS: A Dataset for Large Vocabulary Instance Segmentation*. 2019. arXiv: 1908.03195 [cs.CV].

[6] Bryan A. Plummer et al. *Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models*. 2016. arXiv: 1505.04870 [cs.CV].

[7] Yanbei Chen et al. *Semi-Supervised and Unsupervised Deep Visual Learning: A Survey*. 2022. arXiv: 2208.11296 [cs.CV].

[8] Xingyi Zhou et al. *Detecting Twenty-thousand Classes using Image-level Supervision*. 2022. arXiv: 2201.02605 [cs.CV].

[9] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. *Probabilistic two-stage detection*. 2021. arXiv: 2103.07461 [cs.CV].

[10] Xudong Wang et al. *Cut and Learn for Unsupervised Object Detection and Instance Segmentation*. 2023. arXiv: 2301.11320 [cs.CV].

[11] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].

[12] Zhaowei Cai and Nuno Vasconcelos. *Cascade R-CNN: Delving into High Quality Object Detection*. 2017. arXiv: 1712.00726 [cs.CV].

[13] Xiaoyan Li et al. *Weakly Supervised Object Detection with Segmentation Collaboration*. 2019. arXiv: 1904.00551 [cs.CV].

[14] Ke Yang, Dongsheng Li, and Yong Dou. *Towards Precise End-to-end Weakly Supervised Object Detection Network*. 2019. arXiv: 1911.12148 [cs.CV].

[15] Nicolas Gonthier, Saïd Ladjal, and Yann Gousseau. "Multiple instance learning on deep features for weakly supervised object detection with extreme domain shifts". In: *Computer Vision and Image Understanding* 214 (Jan. 2022), p. 103299. DOI: 10.1016/j.cviu.2021.103299. URL: https://doi.org/10.1016%2Fj.cviu.2021.103299.

[16] Yongkui Shen et al. "Enabling Deep Residual Networks for Weakly Supervised Object Detection". In: Nov. 2020, pp. 118–136. ISBN: 978-3-030-58597-6. DOI: 10.1007/978-3-030-58598-3_8.

[17] Hakan Bilen and Andrea Vedaldi. *Weakly Supervised Deep Detection Networks*. 2016. arXiv: 1511.02853 [cs.CV].

[18] Peng Tang et al. *Multiple Instance Detection Network with Online Instance Classifier Refinement*. 2017. arXiv: 1704.00138 [cs.CV].

[19] Peng Tang et al. *PCL: Proposal Cluster Learning for Weakly Supervised Object Detection*. 2018. arXiv: 1807.03342 [cs.CV].

[20] Jordi Pont-Tuset et al. "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.1 (Jan. 2017), pp. 128–140. DOI: 10.1109/tpami.2016.2537320. URL: https://doi.org/10.1109%2Ftpami.2016.2537320.

[21] Jasper Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104 (Sept. 2013), pp. 154–171. DOI: 10.1007/s11263-013-0620-5.

[22] Bowen Dong et al. *Boosting Weakly Supervised Object Detection via Learning Bounding Box Adjusters*. 2021. arXiv: 2108.01499 [cs.CV].

[23] Shijie Fang et al. *WSSOD: A New Pipeline for Weakly- and Semi-Supervised Object Detection*. 2021. arXiv: 2105.11293 [cs.CV].

[24] Yan Li et al. *Mixed Supervised Object Detection with Robust Objectness Transfer*. 2019. arXiv: 1802.09778 [cs.CV].

[25] Yan Liu et al. *Mixed Supervised Object Detection by Transferring Mask Prior and Semantic Similarity*. 2021. arXiv: 2110.14191 [cs.CV].

[26] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. *Revisiting knowledge transfer for training object class detectors*. 2018. arXiv: 1708.06128 [cs.CV].

[27] Ziang Yan et al. *Weakly- and Semi-Supervised Object Detection with Expectation-Maximization Algorithm*. 2017. arXiv: 1702.08740 [cs.CV].

[28] Yuanyi Zhong et al. *Boosting Weakly Supervised Object Detection with Progressive Knowledge Transfer*. 2020. arXiv: 2007.07986 [cs.CV].

[29] Joseph Redmon and Ali Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016. arXiv: 1612.08242 [cs.CV].

[30] Vignesh Ramanathan, Rui Wang, and Dhruv Mahajan. "DLWL: Improving Detection for Lowshot Classes With Weakly Labelled Data". In: June 2020, pp. 9339–9349. DOI: 10.1109/CVPR42600.2020.00936.

[31] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: 2002.05709 [cs.LG].

[32] Kaiming He et al. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2020. arXiv: 1911.05722 [cs.CV].

[33] Ishan Misra and Laurens van der Maaten. *Self-Supervised Learning of Pretext-Invariant Representations*. 2019. arXiv: 1912.01991 [cs.CV].

[34] Zhirong Wu et al. *Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination*. 2018. arXiv: 1805.01978 [cs.CV].

[35] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV].

[36] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2023. arXiv: 2304.07193 [cs.CV].

[37] Zhiliang Peng et al. *A Unified View of Masked Image Modeling*. 2022. arXiv: 2210.10615 [cs.CV].

[38] Hangbo Bao et al. *BEiT: BERT Pre-Training of Image Transformers*. 2022. arXiv: 2106.08254 [cs.CV].

[39] Hugo Touvron et al. *Training data-efficient image transformers distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV].

[40] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[41] Xinlei Chen and Kaiming He. *Exploring Simple Siamese Representation Learning*. 2020. arXiv: 2011.10566 [cs.CV].

[42] Jean-Bastien Grill et al. *Bootstrap your own latent: A new approach to self-supervised Learning*. 2020. arXiv: 2006.07733 [cs.LG].

[43] Minsu Cho et al. *Unsupervised Object Discovery and Localization in the Wild: Part-based Matching with Bottom-up Region Proposals*. 2015. arXiv: 1501.06170 [cs.CV].

[44] Oriane Siméoni et al. *Localizing Objects with Self-Supervised Transformers and no Labels*. 2021. arXiv: 2109.14279 [cs.CV].

[45] Yangtao Wang et al. *TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut*. 2022. arXiv: 2209.00383 [cs.CV].

[46] Shilong Liu et al. *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection*. 2023. arXiv: 2303.05499 [cs.CV].

[47] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506.01497 [cs.CV].

[48] Olga Russakovsky et al. *ImageNet Large Scale Visual Recognition Challenge*. 2015. arXiv: 1409.0575 [cs.CV].

[49] Piyush Sharma et al. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 2556–2565. DOI: 10.18653/v1/P18-1238. URL: https://aclanthology.org/P18-1238.

[50] Faqiang Wang et al. *A Variational Image Segmentation Model based on Normalized Cut with Adaptive Similarity and Spatial Regularization*. 2020. arXiv: 1806.01977 [cs.CV].

[51] Jianbo Shi and J. Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. DOI: 10.1109/34.868688.