

# Annotation-efficient learning for object discovery and detection

Huy V. Vo

▶ **To cite this version:**

Huy V. Vo. Annotation-efficient learning for object discovery and detection. Artificial Intelligence [cs.AI]. Université Paris sciences et lettres, 2022. English. ⟨NNT : 2022UPSLE124⟩. ⟨tel-03919952v2⟩

**HAL Id: tel-03919952**

**https://hal.science/tel-03919952v2**

Submitted on 11 Apr 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à École Normale Supérieure

# Annotation-efficient learning for object discovery and detection


Soutenue par
**Huy V. Vo**
Le 28 novembre 2022

École doctorale n°386
**Sciences Mathématiques de Paris-Centre**

Spécialité
**Informatique**

Préparée au
Willow - Valeo.ai


Composition du jury :

| | |
|---|---|
| Yannis AVRITHIS<br>Athena Research Centern | *Président du jury*<br>*Examinateur* |
| Tinne TUYTELAARS<br>KU Leuven | *Rapporteur* |
| Andrew ZISSERMAN<br>University of Oxford | *Rapporteur* |
| Cordelia SCHMID<br>Inria and DI/ENS | *Examinateur* |
| Patrick PÉREZ<br>Valeo.ai | *Examinateur* |
| Jean PONCE<br>Inria and DI/ENS | *Directeur de thèse* |

ENS | PSL | Inria INVENTEURS DU MONDE NUMÉRIQUE | valeo.ai | PR[AI]RIE PaRis Artificial Intelligence Research InstitutE

# Acknowledgements

I would like to first thank the members of the thesis committee for reviewing and examining my dissertation. Their constructive comments and questions not only help me understand deeper certain aspects of my work but also see beyond them.

I would like to show my deepest gratitude to my supervisors, Jean Ponce and Patrick Pérez, for their support during the four years. I have known them since I started doing research. Both of them have shaped my young career.

To all my collaborators, I am grateful for all the work we have done together. I have learnt a lot from our discussions. In particular, I would like to thank Elena Sizikova and Oriane Siméoni for being always available for discussions and for their counsel on various topics.

To all my lab-mates and friends, I would like to thank you for all the fun that has helped me overcome numerous stressful deadlines.

Last but not least, I would like to thank my family in Vietnam and my wife for their unconditional support. I hope I made them proud.

# Résumé

Les modèles de détection d'objets dans les images sont des composants importants de systèmes intelligents comme les véhicules autonomes ou les robots. Ils sont typiquement obtenus par l'apprentissage supervisé, ce qui nécessite de grands jeux de données annotées à la main. La construction de tels jeux de données est pourtant coûteuse en temps et en argent, ce qui limite souvent leur taille et leur diversité et, par conséquent, restreint l'applicabilité des détecteurs d'objets. Afin d'éviter ces limitations, des alternatives qui demandent moins de données annotées pour la détection d'objets ont été proposées, comprenant l'apprentissage semi-supervisé, faiblement supervisé, actif ou non-supervisé. L'objectif de cette thèse est de développer de telles méthodes. En particulier, nous nous concentrons sur le problème de découverte d'objets non-supervisée (UOD) et une combinaison de l'apprentissage faiblement supervisé et actif pour la détection d'objets.

Étant donné une collection d'images, la découverte d'objets non-supervisée vise à trouver les images qui contiennent les objets de la même catégorie, et localiser ces objets. Dans la première partie de la thèse, nous proposons quatre approches – OSD, rOSD, LOD et LOST – pour résoudre ce problème. Ces méthodes améliorent graduellement l'efficacité et l'applicabilité de l'UOD.

OSD et rOSD supposent qu'il existe une structure de graphe dans les collections d'images où celles-ci sont les nœuds et deux images sont connectées si elles contiennent des objets d'une même catégorie. Elles reformulent l'UOD comme un problème d'optimisation discrète où les variables binaires décrivent la structure du graphe et les propositions de régions des images. Par rapport à OSD, rOSD introduit des modifications qui réduisent le coût de calcul et améliorent la performance. Différente d'OSD et rOSD, LOD formule l'UOD comme un problème de classement dans le graphe dont les nœuds sont les propositions de régions. Cela permet d'utiliser les méthodes de classement existantes pour trouver des noeuds bien connectés dans les graphes comme PageRank [Page, 1999]. Ces méthodes sont hautement efficaces et parallélisables, et permettent d'appliquer l'UOD à des jeux de données très grands. Finalement, LOST ne considère pas de relation entre les images. Elle se base sur la puissance des descripteurs des transformers auto-supervisés [Caron, 2021] et propose une procédure simple pour trouver un seul objet dans l'image. Puis, elle se sert des objets trouvés comme pseudo annotation pour entraîner des détecteurs d'objets qui sont capables de lier les images similaires et trouver plus d'objets par image.

Il est important d'investiguer les capacités des méthodes non-supervisées mais, dans la pratique, nous avons souvent accès à certaines sources de supervision. Nous considérons dans la deuxième partie de la thèse un scénario pratique pour entraîner un détecteur d'objets où toutes

les images d'entraînement possèdent une annotation faible (les catégories de ses objets) et un petit budget d'annotation additionnel est disponible. Nous entraînons d'abord un détecteur avec les annotations faibles. Puis, nous nous servons du budget additionnel pour annoter un petit nombre d'images d'entraînement qui sont choisies avec les stratégies d'apprentissage actif avec les boites englobantes. Nous peaufinons finalement le détecteur avec toutes les annotations disponibles.

En particulier, nous proposons BiB, une stratégie d'apprentissage actif qui choisit un ensemble divers des images où le détecteur fait le plus d'erreurs. Nous montrons que BiB surpasse toutes les stratégies d'apprentissage actif conventionnelles. Notre méthode améliore significativement la performance du détecteur faiblement supervisé avec seulement un petit coût d'annotation additionnel (1-10 images par classes). Elle démontre alors un meilleur compromis entre la performance de détection et le coût d'annotation que l'apprentissage faiblement et complètement supervisé.

---

**Mots clés :** découvert d'objets, détection d'objets, apprentissage non-supervisé, apprentissage actif, apprentissage faiblement supervisé, optimisation.

# Abstract

Object detectors are important components of intelligent systems such as autonomous vehicles or robots. They are typically obtained with fully-supervised training, which requires large manually annotated datasets whose construction is time-consuming and costly. This thesis studies alternatives to fully-supervised object detection that work with less or even no manual annotation. In particular, we focus on the problem of unsupervised object discovery (UOD), and a combination of active and weakly-supervised learning for object detection.

Given an image collection without manual annotation, unsupervised object discovery aims at identifying pairs of images that contain similar objects and localizing these objects. This is a challenging problem due to the absence of annotation and ambiguities in object definition. We discuss in the first part of this thesis several methods to discern these ambiguities and overcome the challenges, in increasing effectiveness and scalability: OSD, rOSD, LOD and LOST.

In OSD, we define objects as visual patterns that appear frequently in the image collection, and formulate UOD as a discrete optimization problem over a set of binary variables that describe the relation between images and objects in them. An approximate solution to this problem is obtained either by solving a convex relaxed problem or applying a greedy block-coordinate ascent procedure. rOSD extends OSD, introducing several modifications that reduce computational cost and diversify the returned regions. Consequently, it enables the effective discovery of multiple objects per image and the application of UOD on larger datasets. Different from OSD and rOSD, LOD reformulates UOD as a ranking problem based on the analogy between repetitive appearing visual patterns in the image collection and well-connected nodes in a graph of region proposals. This allows the application of existing ranking methods to find well-connected nodes in graphs that are highly efficient and parallelizable such as PageRank [Page, 1999], enabling effective UOD on very large datasets. Finally, LOST does not consider inter-image similarity. It relies instead on a simple seed-growing method that exploits features from recent powerful self-supervised transformers [Caron, 2021] to discover one object in each image. It then uses the discovered objects as pseudo annotation to train object detectors, in a class-agnostic or class-aware fashion, that are able to link similar images together and discover more than one object per image.

Exploring the capacities of unsupervised methods is important, but in practice, we often have access to certain sources of annotation. We consider in the second part of this thesis a practical scenario for training an object detector when all training images have weak annotation (class labels) and an additional small budget for annotation is available. We propose to first train an object detector with the weak annotation. We then use the budget to annotate a small number of images that are carefully selected by an active learning strategy with bounding boxes.

Finally, we fine-tune the detectors with all available annotation. This process is repeated several times to gradually improve the detector.

In particular, we introduce BiB, an active learning technique that is designed to target known confusions of weakly-supervised object detectors, e.g., detecting object parts instead of objects or grouping nearby objects together. We first find images on which such confusions occur, then select a diverse set from those selected to be fully annotated. We show that fine-tuning weakly-supervised object detectors with full annotation on a few images chosen with BiB improves their performance, and reduces significantly the performance gap with fully-supervised object detectors. This demonstrates that our proposed pipeline offers a better trade-off between annotation cost and effectiveness than both weakly- and fully-supervised object detection.

---

**Keywords :** object discovery, object detection, unsupervised learning, weakly-supervised learning, active learning, optimization.

# Table of Contents

**Acknowledgements** i

**Résumé** ii

**Abstract** iv

**Table of Contents** v

**List of Figures** vii

**List of Tables** viii

**1 Introduction** 1
- 1.1 Context . . . 1
- 1.2 Unsupervised Object Discovery . . . 7
  - 1.2.1 Motivation . . . 7
  - 1.2.2 Contributions . . . 9
  - 1.2.3 Related Work . . . 12
- 1.3 Active learning strategies for Weakly-Supervised Object Detection . . . 14
  - 1.3.1 Motivation . . . 14
  - 1.3.2 Contributions . . . 15
  - 1.3.3 Related Work . . . 16

**Part I Unsupervised Object Discovery** 19

**2 Unsupervised Image Matching and Object Discovery as Optimization** 21
- 2.1 Introduction . . . 22
- 2.2 Proposed Approach . . . 23
  - 2.2.1 Problem Statement . . . 23
  - 2.2.2 Relaxing the Problem . . . 24
  - 2.2.3 Solving the Dual Problem . . . 25
  - 2.2.4 Solving the Primal Problem . . . 25
  - 2.2.5 Rounding the Solution and Greedy Ascent . . . 26
  - 2.2.6 Ensemble Post Processing . . . 26
- 2.3 Similarity Model . . . 28
  - 2.3.1 Confidence Score . . . 28
  - 2.3.2 Stand-out Score . . . 29
- 2.4 Experiments and Results . . . 30

2.5 Conclusion and Limitations . . . 36

**3 Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections 37**
3.1 Introduction . . . 38
3.2 Related Work . . . 40
3.3 Proposed Approach . . . 41
3.3.1 Region Proposals from CNN Features . . . 41
3.3.2 Regularized OSD . . . 42
3.3.3 Large-Scale Object Discovery . . . 43
3.4 Experiments . . . 46
3.4.1 Experimental Setup . . . 46
3.4.2 Implementation Details . . . 46
3.4.3 Region Proposal Evaluation . . . 48
3.4.4 Object Discovery Performance . . . 49
3.5 Conclusion and Limitations . . . 54

**4 Large-Scale Unsupervised Object Discovery 56**
4.1 Introduction . . . 57
4.2 Problem Statement and Related Work . . . 59
4.2.1 Problem Statement . . . 59
4.2.2 Related Work . . . 60
4.3 Proposed Approach . . . 60
4.3.1 Quadratic Formulation . . . 60
4.3.2 PageRank Formulation . . . 62
4.3.3 Using (Q) to Personalize PageRank . . . 62
4.4 Experimental Analysis . . . 63
4.4.1 Large-scale Object Discovery . . . 65
4.4.2 Category Discovery . . . 70
4.4.3 Discussions . . . 73
4.5 Conclusion and Future Work . . . 74

**5 Localizing Objects with Self-Supervised Transformers and no Labels 75**
5.1 Introduction . . . 76
5.2 Related Work . . . 77
5.3 Proposed Approach . . . 78
5.3.1 Transformers for Vision . . . 78
5.3.2 Finding Objects with LOST . . . 79
5.3.3 Towards Unsupervised Object Detection . . . 81
5.4 Experiments . . . 82
5.4.1 Experimental Setup . . . 82
5.4.2 Single-Object Discovery . . . 83
5.4.3 Unsupervised Object Detection . . . 84
5.4.4 Ablation Study . . . 87
5.5 Conclusion, Limitations and Future Work . . . 90

**Part II Annotation-efficient object detection 93**

**6 Active Learning Strategies for Weakly-Supervised Object Detection 95**

6.1 Introduction . . . 96
6.2 Proposed Approach . . . 97
6.2.1 Problem Statement . . . 97
6.2.2 Active Learning for Weakly-Supervised Learning Object Detection . . . 98
6.2.3 BiB: An Active Learning Strategy . . . 98
6.2.4 Training Detectors with both Weak and Strong Supervision . . . 101
6.3 Experimental Analysis . . . 102
6.3.1 Experimental Setup . . . 102
6.3.2 Experimental Results . . . 105
6.3.3 Additional Analysis . . . 109
6.4 Conclusion and Future Work . . . 110

**7 Conclusions** **113**
7.1 Contributions . . . 113
7.2 Future Work . . . 115

**List of Publications** **117**

**Appendix** **119**
A Unsupervised Image Matching and Object Discovery as Optimization . . . 119
B Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections . 120
C Large-Scale Unsupervised Object Discovery . . . 124
D Localizing Objects with Self-Supervised Transformers and no Labels . . . 127
E Active Learning Strategies for Weakly-Supervised Object Detection . . . 130

# List of Figures


1.1 Valeo Drive4U is the first autonomous car to drive in the crowded streets of Paris. 2
1.2 An illustration of typical image annotations: (a) class labels, (b) bounding boxes and (c) segmentation masks. . . . . 4
1.3 An illustration of the similarity scores between region proposals in two different images computed with the PHM algorithm [Cho, 2015]. In the pictures, the score of a region is its maximum similarity score with a region in the other image. High scores are show in red and low scores are in blue. The confidence score (second column) often highlights small parts of objects (wheels). It is adjusted (third column) to favor whole-object regions that stand out from the background. . . . 8
1.4 The implicit graph of images that we exploit in OSD (Chapter 2) and rOSD (Chapter 3). In this graph, nodes are images and two images are neighbors if they contain similar objects. . . . . 10
1.5 We consider a graph of regions in LOD (Chapter 4). In this graph, nodes are region proposals generated by off-the-shelf methods and edges between every pair of regions are weighted by their similarity score. In the figure, thicker edges means higher weights. . . . . 11

2.1 The proposed optimization-based method automatically discovers links between images that depict similar objects. This figure shows two image clusters that emerge as a by-product of this approach on the VOC_6x2 object recognition dataset that mixes 6 classes under two viewpoints. See Section 2.4 for details. . . 22
2.2 Visualization of VOC_6x2 in the mixed setting. The figure shows the third component in the graph of regions, corresponding roughly to class *motorbike*. The two first components are shown in Figure 2.1. . . . . 35

3.1 An overview of the different modules of the proposed rOSD method. Given an image collection, we first extract CNN features and generate region proposals for each image (blue zone). These proposals can be divided into disjoint groups, each corresponding to at most one object and shown with a different color. We then run rOSD directly on the collection if it is not too large (orange zone) or run the two-stage large-scale algorithm (red zone) otherwise. . . . . 38
3.2 Illustration of the unsupervised region proposal generation process. The top row shows the original image, the global saliency map $s_g$, local maxima of $s_g$ and three local saliency maps $s_y$ from three local maxima (marked by red stars). The next three rows illustrate the proposal generation process on the local saliency maps: From left to right, we show in green the connected component formed by pixels with saliency above decreasing thresholds and, in red, the corresponding region proposals. . . . . 41

3.3 Regions returned by OSD and rOSD. In each column from top to bottom: original image, image with regions returned by OSD, image with regions returned by rOSD. OSD tends to returns nearly identical regions around a single object while rOSD selects a more diverse set of regions. . . . . 43
3.4 An illustration of the proposed two-stage algorithm for large-scale unsupervised object discovery. We first run proxy rOSD in parallel on different parts of an partition of the image collection to select approximately $\nu$ good regions for each image. In the second stage, we then run rOSD on the entire image set using only the selected region proposals. . . . . 45
3.5 An illustration of persistence in the 1D case. Left: A 1D function. Right: Its persistence diagram. Points above the diagonal correspond to its local maxima and the vertical distance from these points to the diagonal is their persistence. Local maxima with higher persistence are more robust: $B$ is more robust than $A$ although $f(A) > f(B)$. Given a chosen persistence threshold (shown by dash lines in blue), points with persistence higher than some threshold are selected as robust local maxima. The black horizontal dotted lines show birth and death time of the local maxima of $f$. . . . . 47
3.6 Quality of proposals by different methods. (a-c): Detection rate by number of proposals at different $IoU$ thresholds of randomized Prim (RP) [Manen, 2013], edgeboxes (EB) [Zitnick, 2014], selective search (SS) [Uijlings, 2013] and ours; (d): Percentage of positive proposals for the four methods. . . . . 49
3.7 Qualitative multi-object discovery results obtained with rOSD. White boxes are ground-truth objects and red ones are our predictions. Original images are in the first row. Results with $\nu = 50$ and $IoU = 0.7$ are in the second row. Results with $\nu = 25$ and $IoU = 0.3$ are in the third row. . . . . 52

4.1 Sample unsupervised object discovery results obtained by LOD on the OpenImages dataset [Krasin, 2017] which contains 1.7M images. Ground-truth boxes are shown in yellow, and predictions are in red. Best viewed in color. . . . . 57
4.2 An overview of the proposed method LOD. It receives as input a collection of images, each equiped with a set of region proposals. It then builds an undirected weighted graph where nodes are regions and edge weights reflect node similarity. Ranking methods are then applied to obtain for each node-region a score. Finally, the regions with highest scores in each image are returned as objects. . . . . 58
4.3 Comparison of run time as a function of the number of input images. LOD achieves significant improvement in performance and/or savings in run time compared to previous works. [Zitnick, 2014] and [Wei, 2019] are linear in the number of images but their run time are very small compared to other methods and look flat in the figure. . . . . 66
4.4 Examples where our method (LOD) succeeds (left) and fails (right) to discover ground-truth objects in the Op1.7M dataset [Krasin, 2017]. Ground-truth objects are in yellow, our predictions are in red. Best viewed in color. . . . . 67
4.5 Comparison to prior work on the image neighbor retrieval task using CorRet measure. Higher is better. . . . . 71
4.6 Confusion matrix revealing links between the object classes and the clusters found by LOD on VOC_all. . . . . 72

4.7 Performance of LOD by object category and category frequency (number of object occurrences of each category) on the C20K dataset. Results are reported with odAP50, higher numbers are better. Object categories are indexed in decreasing order of category frequency. Performance of LOD is not well-correlated (correlation −0.09) with category frequency. . . . . 73

5.1 Three applications of LOST to unsupervised single-object discovery (left), multi-object discovery (middle) and object detection (right). In the latter case, objects discovered by LOST are grouped into clusters, and cluster indices are used to train a classical object detector. Although large image collections are used as pseudo labels to train the underlying image representation [Caron, 2021], *no annotation* is ever used in the pipeline. See Tables 5.1 and 5.3, and Figure 5.3 for more experiments. . . . . 76

5.2 Initial seed, patch similarities and patch degrees. Each column corresponds to one image. The top row shows original images from the Pascal VOC2007 dataset. The images in the middle row illustrate the initial seed $p^*$ (in red) and patches similar to $p^*$ (in grey), *i.e.*, patches $q$ such that $\mathbf{f}_{p^*}^\top \mathbf{f}_q \geq 0$. The bottom row shows maps of the inverse degree $1/d_p$ of all patches $p$ (yellow corresponds to high inverse degree and blue corresponds to low inverse degree). The initial seed $p^*$ is the patch with the lowest degree. Best viewed in color. . . . . 79

5.3 Objects discovered by LOST on VOC07. The red square represents the seed $p^*$, the yellow box is the box obtained using only $p^*$, and the purple box is the one obtained using all the seeds in $\mathcal{S}$, collected via seed expandion. Using only the initial seed $p^*$, the returned boxes tend to focus only on the most discriminative parts of objects. Seed expansion allows returning larger regions that cover the entire object extent. . . . . 81

5.4 Examples of predictions obtained by the class-aware detector LOST + OD on COCO (a different color per class). The actual "person" class is assigned three different pseudo-classes, illustrating the difficulty to "see" a single category for a "person" in very different configurations. . . . . 85

5.5 An illustration of the effect of seed expansion on VOC07. In each image, the red square represents the seed $p^*$, the yellow box is obtained using only $p^*$, and the purple box is obtained using all the seeds $\mathcal{S}$ with $k = 100$. . . . . 89

5.6 Failure cases of seed expansion on VOC07. In each image, the red square represents the seed $p^*$, the yellow box is obtained using only $p^*$, and the purple box is obtained using all the seeds $\mathcal{S}$ with $k = 100$. . . . . 90

6.1 Overview of our approach. A base weakly-supervised object detector is first trained with image-level tags only, then fine-tuned in successive stages using few *well-selected* images that are fully-annotated. For their selection, we propose *"box-in-box"* (BiB), an acquisition function designed to discover recurring failure cases of the weakly-supervised detector, e.g., failure to localize whole objects or to separate distinct instances of the same class. . . . . 96

6.2 Example of box-in-box (BiB) pairs among the predictions of the weakly-supervised object detector. The existence of such pairs is an indicator of the detector's failure on those images. In the images, boxes of different colors are prections of different classes and the numbers represent the prediction confidence. Best viewed in color. 99

6.3 Detection performances in AP50 of different active learning strategies in our framework on VOC07 [Everingham, 2007] (a) and COCO datasets [Lin, 2014] (b). We perform 5 annotation cycles for each strategy with the budget of $B = 50$ images per cycle on VOC07 and $B = 160$ images per cycle on COCO. This corresponds to annotating 1% and 0.2% of the training set per cycle respectively for the two datasets. Dashed lines in purple and red highlight results obtained with *10-shot* and *10%* images selected with *u-random*. Best viewed in color. . . . . 106
6.4 Images selected by BiB, *entropy-max* and *loss* strategies on COCO dataset. Images selected by *loss* tend to depict complex scenes, many of which are indoors scenes with lots of objects (people, food, furnitures, ...). The supervision brought by these images is both redundant (too many images for certain classes) and insufficient (no or too few images for others). *entropy-max* tends to select very difficult images that are not representative of the training dataset. In contrast, BiB selects a diverse set of images that reflect the detector's confusion on object extent. As a result, BiB significantly outperforms the others on this dataset. . . . 107
6.5 Examples of predictions on the VOC07 and COCO test sets, by MIST [Ren, 2020a] (first row) and BiB after the first cycle (second row). Fine-tuning MIST with images selected by BiB significantly remedies its limitations. . . . . 109

B.1 Multi-object discovery performance of rOSD compared to OSD and [Wei, 2019] when varying the maximum number of returned objects. . . . . 121
B.2 Multi-object discovery results. In each column, from top to bottom: original image, image with predictions of OSD, image with predictions of rOSD. White boxes are ground-truth objects and red ones are our predictions. There are *at most* 5 predictions per image. . . . . 122
C.1 Examples in the COCO [Lin, 2014] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red. . . . . 124
C.2 Examples in the OpenImages [Krasin, 2017] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red. . . . . 124
C.3 Examples in the COCO [Lin, 2014] and OpenImages [Krasin, 2017] datasets where LOD fails to discover ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red. . . . . 125
E.1 Examples of *box-in-box* (BiB) pairs on VOC07 (first two rows) and COCO (last two rows) extracted using the MIST [Ren, 2020a] detector. . . . . 130

# List of Tables


2.1 Performance of different configurations of our algorithm compared to the results of [Cho, 2015] on Object Discovery and VOC_6x2 datasets in the separate setting. Best results are in bold. We observe that both the normalized score (NS) and the ensemble method (EM) improve the performance. EM also improves the stability of our solution (lower variance). The combination of ensemble method (EM), continuous optimization (CO) and normalized scores (NS) produces the best results for OSD. . . . . 31

2.2 Performance on VOC_all in separate setting with different configurations of our method compared to baselines. The combination of continuous optimization (CO) and ensemble method (EM) yeilds the best results for our method. Note that [Li, 2016] and [Wei, 2017a] use pre-trained CNN features [Simonyan, 2015a] while [Cho, 2015] and our method use the hand-crafted WHO [Hariharan, 2012] features. 32

2.3 Performance of our method compared to [Cho, 2015] in the mixed setting. . . . . 33

2.4 Performance of different configurations of our algorithm with $\nu = 1$, $\nu = 5$ and $\nu = 10$. Larger values of $\nu$ yield better performance but we use $\nu = 5$ in our experiments to facilitate comparisons to [Cho, 2015]. . . . . 33

2.5 Performance of our algorithm with deep features on VOC_6x2 in the separate setting. . . . . 34

2.6 Object discovery on VOC_6x2 with region proposals generated by selective search [Uijlings, 2013] and randomized Prim [Manen, 2013]. . . . . 34

3.1 Left: Colocalization performance with our proposals in different configurations of OSD. Right: Colocalization performance for different values of hyper-parameters. 50

3.2 Single-object colocalization and discovery performance of OSD with different types of proposals. We use VGG16 features [Simonyan, 2015a] to represent regions in these experiments. Best results are in bold, second best results are underlined. 50

3.3 Single-object colocalization performance of our approach compared to the state of the art. Note that [Wei, 2019] outperforms our method on VOC_all and VOC12 in this case, but the situation is clearly reversed in the much more difficult discovery setting, as demonstrated in Table 3.4. OSD$^\dagger$ denotes the original OSD in Chapter 2. . . . . 51

3.4 Single-object discovery performance on the datasets with our proposals compared to the state of the art. OSD$^\dagger$ denotes the original OSD in Chapter 2. . . . . 51

3.5 Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets. OSD$^\dagger$ denotes the original OSD in Chapter 2. . . . . 52

3.6 Performance of our large-scale algorithm compared to the baselines. Our method and baseline 1 have the same memory cost, which is much smaller than the cost of baseline 2 . Also, due to memory limits, we cannot run baseline 2 on C20K. . 53
3.7 Influence of the ensemble method on the colocalization performance of OSD and rOSD with our proposals. . . . . 54
3.8 Influence of the ensemble method on the single-object discovery performance of OSD and rOSD with our proposals. . . . . 54
3.9 Quality of the returned local image graph as measured by CorRet. . . . . 54

4.1 Large-scale object discovery performance and comparison to the state of the art on COCO [Lin, 2014] (C120K), OpenImages [Krasin, 2017] (Op1.7M) and their respective subsets C20K and Op50K, in three standard metrics. Using VGG16 features [Simonyan, 2015a], the proposed method LOD achieves top performance in both single and multi-object discovery, and scales better to 1.7M images in Op1.7M than rOSD. When running with self-supervised features (LOD + Self [Gidaris, 2021]), it yields the best results on Op1.7M, showing the first effective fully unsupervised pipeline for UOD. See Section 4.4 for more details. . . . . 59
4.2 UOD performance with supervised [Simonyan, 2015a] (Sup) and self-supervised [Gidaris, 2021] (Self) features on C20K and Op50K datasets. Region proposals are generated by methods from Edgeboxes[Zitnick, 2014] and rOSD with different types of features. LOD with self-supervised features yields reasonable results compared to supervised features. Variants of our proposed method LOD yield state-of-the-art performance in all settings. . . . . 68
4.3 A comparison of different ranking methods for UOD. LOD is better than (Q) and (P) in most of the cases. . . . . 69
4.4 Influence of the damping factor $\beta$ on PageRank's performance (left) and of the selection factor $\alpha$ on LOD's performance (right) on the C20K and Op50K datasets. 69
4.5 The performance of LOD on C20K and Op50K datasets when varying the number of region proposals per image. Using more regions improves LOD's performance. 70
4.6 LOD performance with VGG [Simonyan, 2015a] and ResNet50 [He, 2016] features on C20K and Op50K datasets. Although the latter are more powerful in image classification, VGG16 features yield the best results in object discovery with LOD. 70
4.7 Purity (↑) of our clustering method compared to the state of the art in category discovery on the SIVAL dataset [Rahmani, 2008]. Following prior work, we perform the task on a partition of the dataset and report the average purity on its parts as the final result. Results of other methods are from [Zhang, 2015b]. . . . 71

5.1 Single-object discovery performance in CorLoc on VOC07 trainval, VOC12 trainval and C20K. We compare LOST to state-of-the-art object discovery methods [Kim, 2009; Wei, 2019; Zhang, 2020, rOSD, LOD] as well as to two object proposal methods [Uijlings, 2013; Zitnick, 2014]. We also compare to the segmentation method proposed in DINO [Caron, 2021], denoted by DINO-seg. Additionally, we train a class-agnostic dectector (+ CAD) using as ground-truth either our pseudo-boxes or the boxes of rOSD or LOD. . . . . 83
5.2 Single-object discovery performance in CorLoc of LOST with features originating from different backbones: ViT [Dosovitskiy, 2021] small (ViT-S) and base (ViT-B) with patch size $P = 8$ or 16, ResNet50 [He, 2016] pre-trained following DINO [Caron, 2021], and VGG16 [Simonyan, 2015a] and ResNet50 trained in a fully-supervised fashion on Imagenet [Deng, 2009]. . . . . 84

5.3 Object detection performance (AP50) on VOC07 test. LOST + OD and rOSD + OD are trained on VOC07 trainval. LOST + OD† is trained on the union of VOC07 and VOC12 trainval sets. . . . . 85
5.4 Class-agnostic unsupervised object detection performance in AP50. Trainings, corresponding to '*method* + CAD', are performed on the unlabelled images and rely only on the fully-unsupervised methods rOSD, LOD and LOST (ours). Evaluation of unsupervised object detection may thus be performed on the same images as those used for unsupervised training (without manual annotations). The classic methods EdgeBoxes [Zitnick, 2014] and Selective Search [Uijlings, 2013] do not involve any training. . . . . 86
5.5 Multi-object discovery performance in odAP (average precision for object discovery) of LOST and the baselines [Kim, 2009; Wei, 2019; rOSD; LOD]. . . . . 86
5.6 Image neighbor retrieval performance (CorRet) of different methods and features. 87
5.7 CorLoc performance on VOC2007 with different choices of transformer features in the seed selection, expansion and box extraction steps, as well as influence on the results of the parameter $k$ (maximum number of patches with the lowest degree, in $\mathcal{D}_k$, for seed expansion). . . . . 88
5.8 Impact of number of clusters in object detection. Results, using the mean AP50 (%) across all the classes, on VOC07 test. All models are trained using LOST's pseudo-boxes (i.e., LOST + OD) on the VOC07 and VOC12 trainval sets. The number of classes in VOC is 20. . . . . 90

6.1 Ablation study. Results in AP50 on VOC07 with 5 cycles and a budget $B = 50$. We provide averages and standard deviation results over 6 repetition. *DifS* stands for the difficulty-aware region sampling module. Images are selected by applying k-means++ init. (*K selection*) on image-level features (*im.*), confident predictions' features (*reg.*) or BiB pairs. . . . . 105
6.2 Performance of BiB compared to the state of the art on VOC07 ($B = 50$) and COCO ($B = 160$) datasets. The *10-shot* setting corresponds to 4 and 5 AL cycles resp. on VOC07 and COCO. All of the compared methods use VGG16 [Simonyan, 2015b] as the backbone. . . . . 108
6.3 Per-class AP50 results on VOC07. BiB yields significant boosts in hard classes such as *bottle*, *chair*, *table* and *potted plant*. Results of MIST [Ren, 2020a] are the average of three runs using the authors' public code and differ from the numbers in the original paper. . . . . 108
6.4 A comparison between BiB, u-rand and two other variants that combine them. BiB outperforms the variants, showing that diversity sampling is important to the effectiveness of BiB. . . . . 109
6.5 Performance of BiB on VOC07 with different values of the area ratio $\mu$ in BiB design. We conducted 5 cycles with a budget of 50 images per cycle, repeated the experiment six times for each value of $\mu$ and report the average and standard deviation of their performance. . . . . 110

B.1 Single-object colocalization performance of our approach compared to the state of the art. Note that Wei *et al.* [Wei, 2019] outperform our method on VOC_all and VOC12 with VGG19 features in this case, but the situation is clearly reversed in the much more difficult single-object discovery setting, as demonstrated in Table B.2. OSD† denotes the original OSD in Chapter 2. . . . . 120

B.2 Single-object discovery performance in the mixed setting on the datasets with our proposals compared to the state of the art. OSD$^\dagger$ denotes the original OSD in Chapter 2. . . . . 120
B.3 Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets. OSD$^\dagger$ denotes the original OSD in Chapter 2. . . . . 121
B.4 Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets when using smaller values of $\nu$ (25) and $IoU$ (0.3) threshold. . . . . 122
B.5 Quality of the returned local image graph as measured by CorRet. . . . . 123
C.1 Large-scale multi-object discovery performance and comparison to the state of the art on COCO [Lin, 2014], OpenImages [Krasin, 2017] and their respective subsets C20K and Op50K, as measured by detection rate. . . . . 125
D.1 DINO-seg ablation study. We compare here CorLoc results on datasets VOC07_trainval, VOC12_trainval and C20K when applying the DINO-seg method to create a box from the different heads of the attention layer. Also, DINO-seg BCC selects the box/head that produces the biggest connected component, and DINO-seg HAIoU selects the box/head that has the highest average IoU with the other 5 boxes. We additionally report results with our method LOST for comparison. . . . . 127
D.2 CorLoc results on the VOC07_noh and VOC12_noh datasets. . . . . 128
D.3 Single-object discovery performance in CorLoc of LOD and LOST with different types of features. . . . . 128
E.1 Comparison of active learning strategies on VOC07. For each experiment, we conducted 5 cycles with a budget of 50 images per cycle. We repeated the experiment six times for each strategy and report the average and standard deviation of their performance (in AP50). BiB yields significantly better performance than the others. *loss* performs well in the first cycle but fares worse than BiB in subsequent cycles. Additionally, it performs much worse, even than random, on COCO (see Table E.2). . . . . 131
E.2 Comparison of active learning strategies on COCO. For each experiment, we conducted 5 cycles with a budget of 160 images per cycle. We repeated the experiment three times for each strategy and report the average and standard deviation of their performance (in AP50 and AP). BiB significantly outperforms all other methods. . . . . 131
E.3 Performance of the loss strategy with different choices of the detector's loss on VOC07. For each experiment, we perform 5 cycles with a budget of 50 images per cycle. We have repeated the experiment six times for each strategy and report the average and standard deviation of their performance. . . . . 132
E.4 Ablation study on COCO. We show the average and standard deviation results over several runs in AP50 on COCO with 5 cycles and a budget $B = 160$. *DifS* stands for the difficulty-aware region sampling module. Images are selected by applying k-means++ init. (*K selection*) on image-level features (*im.*), confident predictions' features (*reg.*) or BiB pairs. . . . . 132

# Chapter 1

# Introduction

## 1.1 Context

Much like the human dreams of flying and exploring far-away worlds in the universe since the dawn of humanity, Artificial Intelligence (AI) has been a fixation in the mind of computer scientists and ordinary people alike since the beginning of the computer age, when the first automatic computer was invented nearly a century ago. Although there are some minor variations in how AI is defined, it is commonly thought as the ability of computers or computer-controlled machines to undertake tasks that are usually done by humans, and at a high level, exercise human characteristics such as thinking, reasoning, planing, imagining, generalizing, etc. With such abilities, a functioning AI, at low or high level, would bring transformational changes to society, from the liberation of humans of ordinary physical tasks, such as household chores, delivery or driving, to enhancing the quality of life through better health care and education services. Such benefits have encouraged many efforts in AI research since the 1950s. Early research brought certains advances which play as groundwork for the field such as Multi-Layer Perceptron [Rosenblatt, 1958], Backpropagation [Kelley, 1960; Bryson, 1961; Dreyfus, 1962; Rumelhart, 1986a; Rumelhart, 1986b] or Convolutional Neural Networks (CNNs) [Fukushima, 1980; LeCun, 1989; Lecun, 1998] but also suffered from the cool down of enthusiasm due to over-hype and the lack of demonstrated practical benefits.

It was not until recently did AI start to find wide-spread applications in practical aspects of life such as automation, health care, customer experience or security. Autonomous vehicles are one of the AI-powered applications that were once futuristic but are coming into reality. It has attracted the attention and investments from car makers, historical or new such as Daimler and Tesla, and global automotive suppliers like Valeo, as well as of tech companies such as Google, numerous start-ups and academic research institutions. Currently at level 3 with a few recent models, in the six levels (0 to 5) defined by the Society of Automotive Engineers, the industry is pushing for the full driving automation (level 5) in the future. As a world leader in automotive sensors, Valeo contributes to this effort, notably with its unique automotive-grade LiDAR, and develops autonomous platforms such as Valeo Drive4U (Figure 1.1), the first autonomous car to drive in the crowded streets of Paris in 2018, and Valeo Cruise4U designed for highway driving, that has taken several long road trips across Europe, Japan and the USA. Health care is another




Figure 1.1 – Valeo Drive4U is the first autonomous car to drive in the crowded streets of Paris.

field revolutionized with AI-powered capacities. In particular, advances in medical imaging have improved the accuracy, effectiveness and efficiency of the diagnosis and treatment of diseases. These advances of AI come from a combination of factors, including more available and faster computational resources [Dean, 2012] and the popularity of personal digital devices that generate the vast amount of data necessary to train large AI models.

Visual perception is an important part of any artificial intelligence system which needs to perceive the surrounding environment in order to interact with it. Computer vision is the scientific and engineering field that tasks itself with enabling computer understanding and interaction with the visual world. Early works in the 1970s focused on extracting the three-dimensional model of visual scenes from two-dimensional images, developing techniques for edge extraction [Roberts, 1963], line labeling [Huffman, 1971; Clowes, 1971; Waltz, 1975; Rosenfeld, 1976; Kanade, 1980] and object representations as an interconnected combination of parts [Fischler, 1973; Hinton, 1977; Marr, 1982]. In the next decades, research on the field moved towards more rigorous mathematical models such as Markov random fields [Geman, 1984; Dickmanns, 1988; Matthies, 1989], regularization [Terzopoulos, 1983; Poggio, 1985; Blake, 1987] or contour models [Kass, 1988], and statistical learning tools were also used to analyze visual data. [Szeliski, 2010] provides a nice brief overview on the early history of Computer vision.

Recently, data-driven, feature-based models that employ machine learning techniques and optimization frameworks have been popular in Computer vision. The most common framework, known as fully-supervised learning, involves building a dataset of input-target pairs, extracting numerical features for the input images then using the features-target pairs to optimize, or "train", a machine learning model. An important block in this pipeline is the feature generation step which produces robust features from images using keypoints or dense histograms of gradients [Lowe, 2004; Dalal, 2005]. More robust features can be obtained by further applying

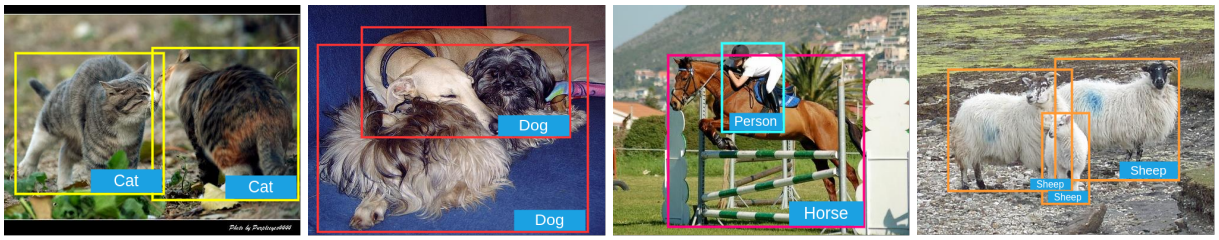
whitening [Hariharan, 2012] or statistical models [Perronnin, 2007; Sanchez, 2013]. These features, called hand-crafted features, are carefully designed to capture key informations for visual perception in images such as edges, corners or blobs. Although great progresses have been made thanks to these efforts, this approach has several limitations. First, designing hand-crafted features is often hard and involved. Second, these features mostly leverage low-level cues while reasoning at a more abstract level is necessary for many perception tasks. Finally, they do not adapt to different data domains and learning tasks due to the separation of the feature extraction and the training stages. Indeed, domain-specific information provides prior knowledge that is helpful in perception and different tasks could need different types of features.

In the last decade, there has been an almost complete shift from this practice to end-to-end learning with neural networks, which perform both feature extraction and prediction. Typically, a neural network consists of a succession of layers, the last of which performs prediction while the others are responsible for feature extraction. During the training of neural networks, the parameters of the feature extractor and the predictor are optimized together in order to match the input data to the expected output. As a result, features are learnt automatically, adapted to the specific learning task and data while also capturing a high level of abstraction. Thanks to its effectiveness, fully-supervised learning with neural networks has become the de facto approach to solving problems in Computer vision.

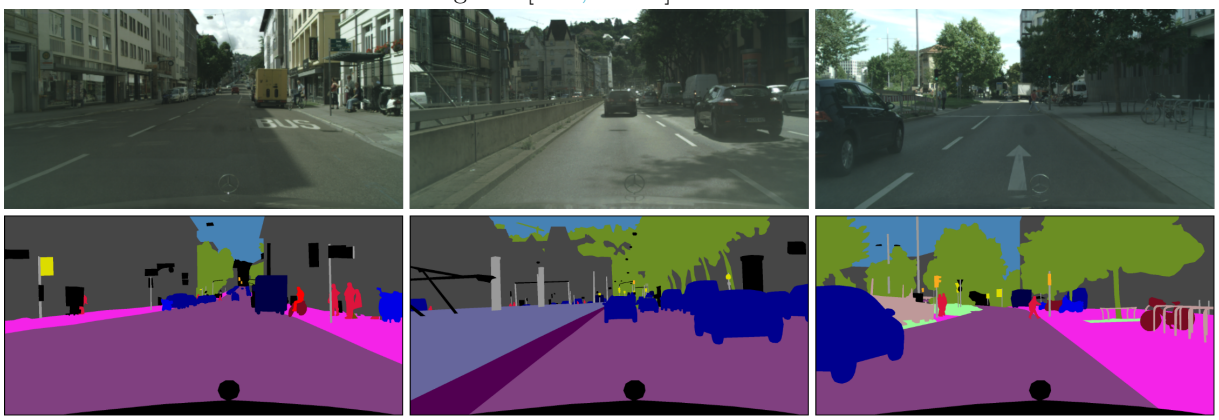
Fully-supervised learning requires a target (label) for each sample in the training dataset. It is, however, not always easy or possible to obtain these labels. Some types of annotations such as those for medical images require a high level of expertise. Others are simply beyond human capacities such as drawing depth maps from 2D images or annotating velocity of vehicles from videos. Even when the acquisition of these annotations is feasible, it is costly both in time and money. For example, drawing a high-quality bounding box – the canonical label for training object detectors – around an object takes approximately 35 seconds [Ren, 2020b], resulting in minutes of annotation time for a single image that depicts multiple objects. The generation of denser annotations such as masks for segmentation tasks takes much longer. For instance, a single image in Cityscapes dataset [Cordts, 2016] requires at least 90 minutes for annotation and quality control of segmentation masks. Typical image annotations are shown in Figure 1.2. The cost of annotation is particularly high when training neural networks. Due to their often large number of parameters and the need to learn the feature extractor and the predictor simultaneously, they require a much larger amount of data than older models such as Support vector machine [Boser, 1992; Cortes, 1995] to be effective. Transfer learning, which consists of reusing the feature extractor from pre-trained neural networks, can partially alleviate this problem but introduces a bias toward the data that are used to pre-train the feature extractor.




(a) Class labels for images in the ImageNet [Deng, 2009] dataset. It takes about 1 second to label one image [Ren, 2020b].




(b) Annotation in forms of bounding boxes on PASCAL VOC2007 [Everingham, 2007] dataset. It takes about 35 seconds to draw one bounding box [Ren, 2020b].



(c) Annotation in forms of segmentation masks on Cityscapes dataset [Cordts, 2016]. It takes more than 90 minutes to annotate each image.

Figure 1.2 – An illustration of typical image annotations: (a) class labels, (b) bounding boxes and (c) segmentation masks.

An effort to find alternatives to the annotation-hungry supervised learning has resulted in a spectrum of other learning paradigms such as semi-supervised, weakly-supervised, self-supervised, low-shot, active and unsupervised learning.

Semi-supervised learning requires target for only a portion of the dataset and leverages information extracted from unlabeled data [Belkin, 2004; Kingma, 2014; Laine, 2017; Tarvainen, 2017; Radosavovic, 2018; Wang, 2018a; Jeong, 2019; Miyato, 2019; Berthelot, 2019; Xie, 2020; Berthelot, 2020; Sohn, 2020a; Li, 2020; Sohn, 2020b; Zoph, 2020; Tang, 2021b; Xu, 2021; Chen,

2021]. Training often involves producing pseudo labels for unannotated data or enforcing the consistency between the model predictions on their different perturbed versions.

Weakly-supervised learning requires a weaker form of target for each input, e.g., image labels instead of bounding boxes for object detection [Bojanowski, 2015; Bilen, 2016; Cinbis, 2017; Jie, 2017a; Tang, 2017; Zeng, 2019; Arun, 2019; Wan, 2019; Gao, 2019b; Ren, 2020a; Huang, 2020; Chen, 2021] or pixel-wise labels for semantic segmentation [Kolesnikov, 2016; Vernaza, 2017; Wei, 2018; Ahn, 2018; Ahn, 2019; Lee, 2019; Wang, 2020b; Zhang, 2020c]. Similar to semi-supervised learning, the model is typically trained using pseudo-labels which, in this case, are generated for all training images by leveraging the weak labels [Kolesnikov, 2016; Tang, 2017; Wei, 2018; Ahn, 2019; Ren, 2020a; Huang, 2020; Zhang, 2020c]. There are also some variants of this seting that combine weak and full supervision [Pan, 2019; Biffi, 2020] or a small amount of weak supervision and a large amount of unannotated data [Yang, 2021].

Low-shot learning aims to leverage fully labelled training data for a task to other unseen, but related tasks which have just a few input-target pairs [FeiFei, 2006; Vinyals, 2016; Snell, 2017; Munkhdalai, 2017; Wang, 2018d; Sung, 2018; Qiao, 2018; Kang, 2019; Fan, 2020; Sun, 2021]. Typical techniques are meta-learning, which involves training a meta-learner that outputs a model for a new task given as input several training examples of the task [Munkhdalai, 2017; Ravi, 2017] or metric learning, which attempts to learn an embedding space whose induced distance approximates well the "sematical distance" between data points [Koch, 2015; Vinyals, 2016; Snell, 2017; Karlinsky, 2019].

Self-supervised learning aims to learn feature extractors that are useful in down-stream tasks in a transfer learning fashion without requiring any target. Popular approaches often use proxy tasks [Doersch, 2015; Pathak, 2016; Zhang, 2016; Noroozi, 2016; Zhang, 2017; Noroozi, 2017; Gidaris, 2018a], contrastive learning [Oord, 2018; Hjelm, 2019; Tian, 2020; Bachman, 2019; Chen, 2020b; He, 2020], clustering [Bojanowski, 2017; Caron, 2018; Caron, 2019], student-teacher models [Grill, 2020; Caron, 2020; Gidaris, 2021; Caron, 2021] or regularization [Zbontar, 2021; Bardes, 2022].

Active learning reduces the annotation cost by selecting and annotating samples that are most relevant to training the model [Settles, 2009; Geifman, 2017; Sener, 2018; Brust, 2019; Zhdanov, 2019; Haussmann, 2020; Agarwal, 2020; Siméoni, 2021a; Yuan, 2021b; Choi, 2021]. In this paradigm, learning typically involves multiple cycles in which the model is trained using the available annotation then used to select new samples for annotation. Samples are typically chosen to encourage diversity [Geifman, 2017; Sener, 2018; Zhdanov, 2019; Agarwal, 2020], to be chalenging to the current model [Gal, 2017; Beluch, 2018; Ducoffe, 2018], or both [Huang, 2014; Hsu, 2015; Ash, 2020].

Unsupervised learning aims to extract useful information from data only, without using any labels [Weber, 2000; Sivic, 2005; Russell, 2006; Grauman, 2006; Arthur, 2007; Kim, 2009; Faktor, 2012; Cho, 2015; Burgess, 2019; Locatello, 2020]. The form of extracted information varies in the literature, e.g., clusters of images that contain similar objects [Weber, 2000; Grauman, 2006; Faktor, 2012], object locations [Sivic, 2005; Russell, 2006; Cho, 2015], pairs of similar images [Cho, 2015] or image decomposition into object masks [Burgess, 2019; Locatello, 2020].

Objects are primitives of visual scenes. In order to comprehend a scene, it is often easier to break it down into objects and analyze each of them individually before considering them collectively. For example, an autonomous driving car would divide its visual field into vehicles, pedestrians, buildings, street lanes, etc.; locate each of these components, analyze their behaviors, movements and interactions before making its decisions. As a result, perception tasks in Computer vision typically concern and are defined after different properties of objects, such as their categories, locations, movements, etc. Object classification aims to group images into different classes based on the categories of objects they contain [Krizhevsky, 2012; Simonyan, 2015a; He, 2016; Huang, 2017; Tan, 2019; Touvron, 2020; Dosovitskiy, 2021]. The design of such object categories depends on the task and can be coarse or fine-grained [Lin, 2015; Guo, 2019; Zhuang, 2020; He, 2021]. Object detection aims to localize and label each object in an image where the object location is described using a tight bounding box enclosing its extent [Girshick, 2014; Gidaris, 2015; Girshick, 2015; Ren, 2015a; Bell, 2016; Redmon, 2016; Redmon, 2017]. Semantic segmentation decomposes the image into different regions, each containing pixels that belong to either an object category of interest or "background" [Long, 2015; Chen, 2018a; Chen, 2018b; Vu, 2019; Strudel, 2021; Xie, 2021a]. Instance segmentation is similar to object detection but returns an object mask instead of a bounding box [He, 2017; Huang, 2019; Chen, 2019b; Chen, 2019a; Fang, 2019; Wang, 2020a; Vu, 2021; Cheng, 2022]. Tracking estimates the object location in subsequent frames given its position in the first frame of a video [Wojke, 2017; Bergmann, 2019; Wang, 2020c; Zhang, 2021]. Some tasks are more specialized on certain classes of objects such as humans [Liu, 2015; Nguyen, 2016; Cao, 2017] or vehicles [Song, 2019; Ouaknine, 2021]. Action recognition focuses on detection and classification of human activities [Wang, 2013; Wang, 2016; Hara, 2018; Wang, 2018b; Radford, 2021]. Trajectory prediction aims to foretell the future movement of vehicles [Lee, 2017; Deo, 2018; Mangalam, 2020; Buhet, 2020]. These tasks require understanding not only individual objects but also their environment and the interactions between them.

When discussing such tasks, it is important to consider the intrinsic ambiguity of an "object": A visual pattern can be considered an object in a scene but not an object in another. In particular, there is often confusion in the distinction between object parts *vs.* objects, objects *vs.* object groups or foreground objects *vs.* background. For example, a wheel standing alone in a scene is seen as an object but often overlooked and considered as an object part when it is figured in a car, which is now considered the object. Also, a banana and a hand of bananas can both be referred to as objects. This ambiguity is often ignored in supervised settings where objects are defined by examples through manually annotated datasets, *i.e.* patterns that are annotated are objects while others are not, even though the latters are considered objects in common sense. This setting restricts the scope of the applications of visual models solely to object classes that are labeled. To have an idea of how limited this setting is, it is interesting to consider Openimages [Krasin, 2017], one of the largest public datasets for object detection. It has only 600 object classes while an ordinary human can easily recognise thousands. The restriction to a few finite lists of categories is mainly due to the costly and time-consuming process of annotation acquisition which involves the participation of human annotators.

In this thesis, we focus on the problem of localizing and discovering objects in images with limited or no supervision. An important part of our work is dedicated to the challenging unsupervised object discovery (UOD) problem and the other part focuses on the study of active learning methods for bridging the gap between weakly- and fully-supervised object detection.

## 1.2 Unsupervised Object Discovery

### 1.2.1 Motivation

Remarkable progress has been achieved in visual tasks such as image categorization, object detection, or semantic segmentation, typically using fully-supervised algorithms and vast amount of manually annotated data [Lazebnik, 2006; Felzenszwalb, 2010; Krizhevsky, 2012; Ren, 2015a; Russakovsky, 2015; He, 2016; He, 2017]. With the advent of crowd-sourcing, large corporations and, to a lesser extent, academic units can launch the corresponding massive annotation efforts for specific projects that may involve millions of images [Russakovsky, 2015]. But handling Internet-scale repositories of images (or videos) or continuous learning scenarios associated with digital assistants or autonomous cars demands approaches less hungry for manual annotation. We have previously discussed several possible alternatives, including weakly-supervised, semi-supervised, self-supervised and active learning. We address in the first part of this thesis the even more challenging problem of unsupervised object discovery. Given a collection of unlabelled images, we aim to discover both the structure of the image collection – that is, which images depict similar objects (or textures, scenes, actions, etc.) – and the objects in question, in a *fully unsupervised* setting [Russell, 2006; Sivic, 2008; Lee, 2010; Faktor, 2012; Rubinstein, 2013; Cho, 2015]. Although weakly-, semi-, and self-supervised methods may provide a more practical foundation for large-scale visual recognition, the fully unsupervised construction of image models is an important and fundamental scientific problem in computer vision. On the one hand, discovering object concepts and locations in an unsupervised fashion, therefore bypassing the need for costly annotation acquisition process, has the potential for leveraging a seemingly unlimited source of image data from the Internet. On the other hand, unsupervised object discovery has a wide range of applications. A direct application is the automatic labeling and organization of large image databases where images containing similar objects of potential interest are linked ahead of time. This would save innumerous hours of human effort and enable more efficient exploitation of these databases in tasks such as interactive query-based visual search. The output of unsupervised object discovery – rough object locations and pairwise image relation – can also be used as noisy annotations which can be leveraged to improve the efficiency of models in related problems such as semi- or weakly-supervised object detection. Moreover, by exploiting the discovered objects and meaningful parts of images, it is possible to improve the training of recent self-supervised learning methods [Mishra, 2021; Hénaff, 2022] which until recently were only focusing on the global context of images. Last but not least, industrial entities such as Valeo spend millions of dollars a year on data annotation for object detection tasks where human operators are asked to label objects and images manually. The cost of labelling campaigns could be cut significantly by presenting discovered boxes, returned

by unsupervised object discovery methods, as candidates for annotators to select before scaling through automatic label propagation.

Unsupervised object discovery is a very challenging task, particularly on natural, in-the-wild images with the presence of occlusions, intra-class variations and background clutter. Also, unlike the supervised object detection/localization problems where objects are "defined" with examples in training data, there is no clear "object" definition in unsupervised object discovery. As a result, the task suffers from the ambiguities of object definitions mentioned earlier. Discovering objects therefore involves eliminating, or rather reducing, these ambiguities. We employ a few tools for this purpose in our approaches.

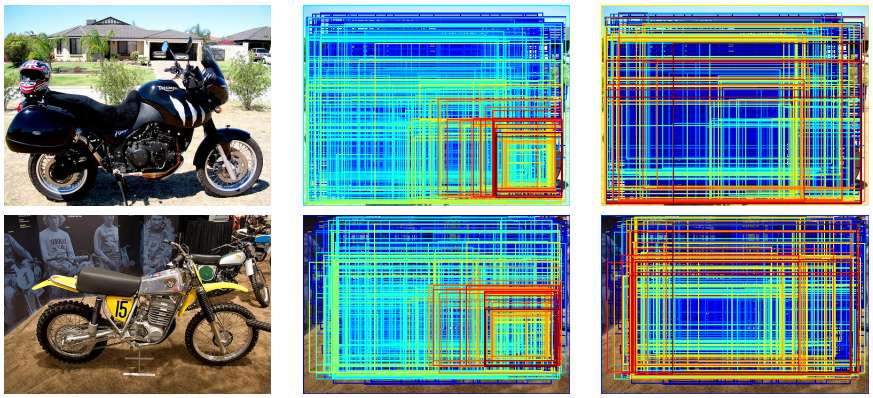


Figure 1.3 – An illustration of the similarity scores between region proposals in two different images computed with the PHM algorithm [Cho, 2015]. In the pictures, the score of a region is its maximum similarity score with a region in the other image. High scores are show in red and low scores are in blue. The confidence score (second column) often highlights small parts of objects (wheels). It is adjusted (third column) to favor whole-object regions that stand out from the background.

First, following [Cho, 2015], we leverage the information from multiple images and define objects as visual patterns that appear frequently in the image collection. Objects are ambiguous in a single image but they can be better defined when appearing in different contexts. For example, it is not clear if a rescue vehicle pulling a car is a single object or two separate objects from a single image but if there are other images that figure cars alone and rescue vehicles alone, we can confidently say it is the latter case. Similarly, sheep are often captured standing close, occluding each other, which makes distinguishing them challenging. In this case, matching different patches of one image to other images that figure individual sheeps could help. With this definition, we propose to find in each image visual patterns (patches) that are *similar* to those present in other images and select them as discovered objects.

Second, we use region proposals, generated by off-the-shelf methods [Uijlings, 2013; Zitnick, 2014] or our own [Vo, 2020] as object priors, and cast finding objects as the selection of several good proposals amongst thousands found in each image. Region proposals are typically generated using low-level cues such as superpixels, edges or color homogeneity, and they generally concentrate around object areas in the images. We exploit the difference in region proposal density, computing the region similarity score with the probabilistic Hough matching (PHM)

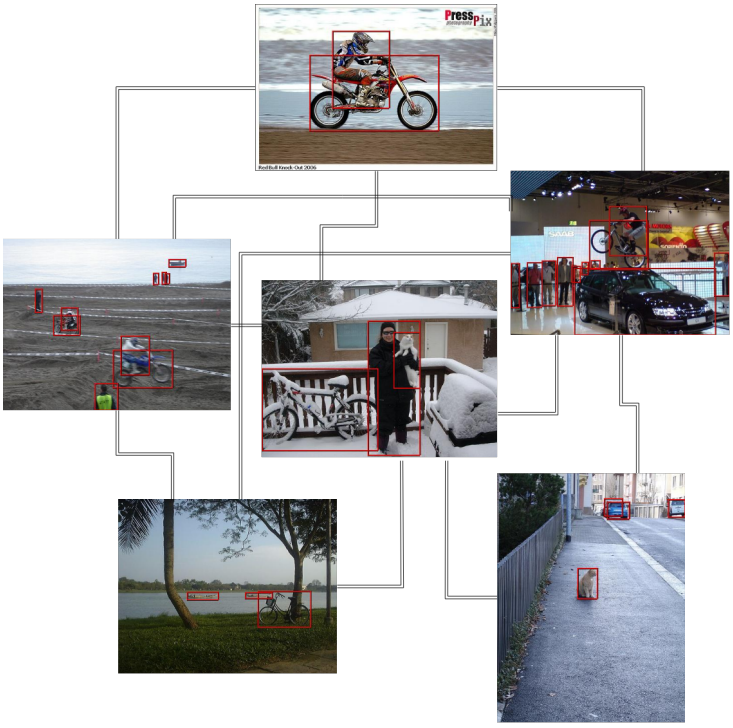


Figure 1.4 – The implicit graph of images that we exploit in OSD (Chapter 2) and rOSD (Chapter 3). In this graph, nodes are images and two images are neighbors if they contain similar objects.

algorithm proposed in [Cho, 2015], to disambiguate between background and foreground. We also adjust the similarity scores computed with the PHM algorithm to favor regions that *stand out* from the background to distinguish objects from object parts. An illustration of region proposals and the PHM similarity scores is shown in Figure 1.3.

Third, we use certain types of features that are trained for other tasks but may capture some information about object location in images during traning to represent region proposals or to directly localize objects. They can be extracted from convolutional neural networks (CNNs) or transformers that are pre-trained for image classification [Simonyan, 2015a; He, 2016] or trained in a self-supervised fashion [Gidaris, 2021; Caron, 2021]. Typically, these features contain information that can be leveraged to distinguish objects from background: CNN features corresponding to objects typically have higher numerical values than those for background while foreground patches and background are well separated in the feature space of self-supervised transformers.

### 1.2.2 Contributions

In this thesis, we discuss several approaches to unsupervised object discovery: Discrete optimization-based OSD [Vo, 2019] and rOSD [Vo, 2020], ranking-based LOD [Vo, 2021] and seed-growing-based LOST [Siméoni, 2021b] [1]. In OSD [Vo, 2019], we consider the implicit graph

1. The code names respectively stand for Object and Structure Discovery, regularized OSD, Large-scale Object Discovery and Localize Objects with Self-supervised Transformers.

structure in image collections where nodes are images, and edges connect two nodes if the corresponding images contain similar objects (Figure 1.4). Based on this graph structure, we formulate unsupervised object discovery as a discrete optimization problem, maximizing the total similarity between objects in neighboring images over the structure of the graph and the choice of good regions to be objects in each image. An approximate solution to this problem can be found with convex optimization techniques and/or a greedy block coordinate ascent procedure. Our following work rOSD [Vo, 2020] improves upon OSD with the introduction of region proposals that possess a nice intrinsic group structure generated from CNN features, a regularized version of the OSD optimization formulation, enabled by the structure of the proposals, and a two-stage algorithm that scales unsupervised object discovery to datasets several times larger than those considered in OSD.

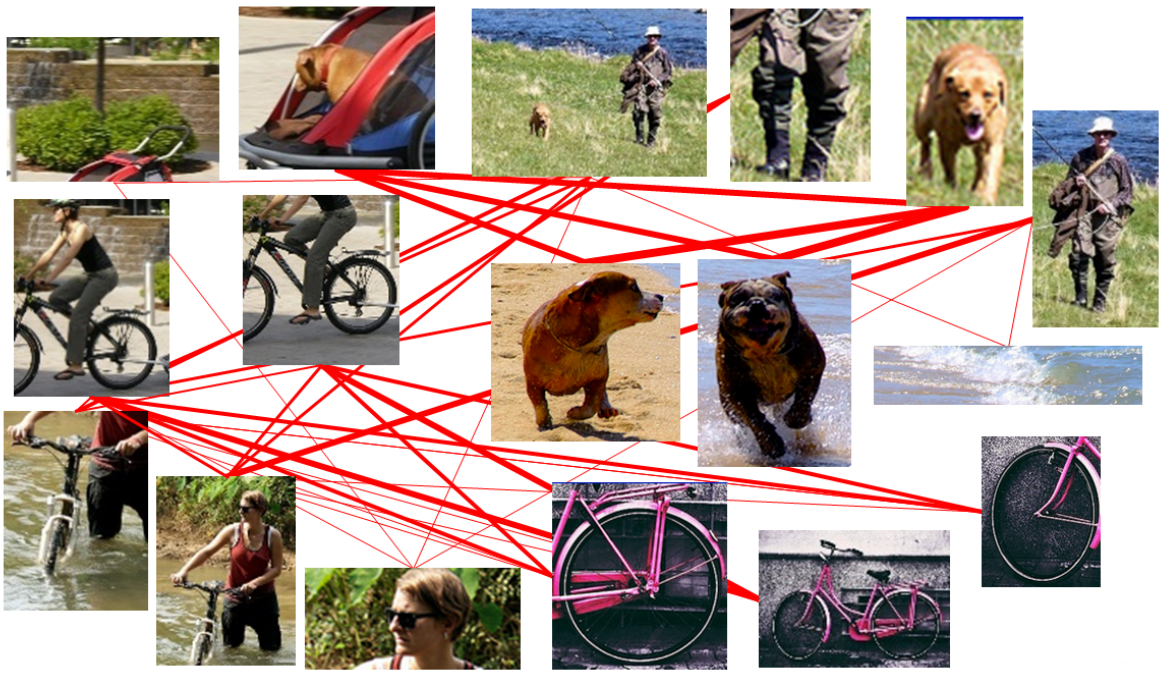


Figure 1.5 – We consider a graph of regions in LOD (Chapter 4). In this graph, nodes are region proposals generated by off-the-shelf methods and edges between every pair of regions are weighted by their similarity score. In the figure, thicker edges means higher weights.

In LOD [Vo, 2021], we observe the parallel between our object definition - visual patterns that appear frequently in the image collections - and well-connected nodes in the complete graph where nodes are regions and edges are weighted by region similarity score (Figure 1.5). Based on this observation, we reformulate unsupervised object discovery as a ranking problem with the goal of assigning a score to each region in the graph such that a higher score means the corresponding region is more likely to be an object. Regions are then selected separately on each image in decreasing score order. This new formulation enjoys efficient solutions that can be implemented in a distributed manner, therefore can leverage large clusters of machines to scale up unsupervised object discovery to very large datasets.

Recent advances in self-supervised feature learning have enabled new possibilities to tackle unsupervised object discovery. In particular, features extracted from DINO [Caron, 2021] have been shown to contain explicit information about image layout and object locations. Our LOST algorithm [Siméoni, 2021b] exploits this property and localizes a single object per image with high precision using a simple seed-growing procedure. We go one step further by using the output of LOST as pseudo labels to train an object detector in a class-agnostic or class-aware fashion. The trained class-agnostic object detector is able to discover multiple objects per image while the class-aware object detector displays performance that is competitive with some weakly-supervised object detectors.

These works have improved the unsupervised object discovery literature in multiple directions. First, we have gradually raised the bar for UOD methods, with subsequent works significantly outperforming their predecessors. Second, we are the first to consider the discovery of multiple objects per image, a more realistic setting on datasets of natural images which often contain various objects of multiple categories. Third, we have gradually improved the efficiency and applicability of unsupervised object discovery methods, increasing the size of the datasets where they are applicable from a few hundreds to several millions. Finally, we successfully investigate the use of pre-trained features from neural networks in unsupervised object discovery which is not trivial and requires special care, as shown in OSD and rOSD.

Finding the optimal evaluation protocol for unsupervised object discovery is not trivial. Annotation is not required to run UOD methods but it is necessary for their evaluation. In practice, we show the effectiveness of our methods on datasets built for object detection such as PASCAL VOC [Everingham, 2007; Everingham, 2012], COCO [Lin, 2014] or OpenImages [Krasin, 2017]. Typically, we compare the object locations discovered by the method to the ground-truth annotation and the model's prediction is counted as a true prediction if its overlap with the ground truths surpasses a certain threshold. This practice is actually common for related tasks such as image colocalization or object co-segmentation [Rubinstein, 2013; Tang, 2014; Li, 2016; Wei, 2019; Hsu, 2019].

In our evaluation of unsupervised object discovery, we consider different settings, each of which needing an adapted evaluation metric. Previous works in UOD and image co-localization consider only the single-object setting where the goal is to find only a single object per image. In this case, the common metric is *Correct Localization* (CorLoc), defined as the percentage of images in which at least one object is correctly localized. In the context of unsupervised object discovery, an object is said to be correctly localized if its *intersection over union* measure with the predicted box is at least 0.5. In most cases, it is desirable to be able to discover multiple objects per image. In this setting, CorLoc is no longer suitable since, according to this metric, a model which is able to discover two objects is considered no better than a model discovering only one object. Following the region proposal literature, one can instead use *detection rate*, or recall, of the predicted boxes when the model returns up to $m$ boxes as the metric. However, this metric depends on the number $m$ which complicates the comparison between different methods. To remedy this issue, we propose a new metric, *object discovery Average Precision* (odAP), which is defined as the area under the precision-recall curve where precision and recall of the

predicted boxes are computed at different values of $m$.

### 1.2.3 Related Work

**Unsupervised object discovery** aims to extract useful information objects in images without any kind of supervision. The concrete form of the extracted information varies in the literature. Early works focus on finding groups of images depicting objects of the same categories, employing probabilistic models [Weber, 2000; Sivic, 2005; Russell, 2006], non-negative matrix factorization (NMF) [Tang, 2008] or clustering techniques [Grauman, 2006], see [Tuytelaars, 2010] for a survey. In addition to finding image groups, some of these approaches, e.g., topic modeling [Russell, 2006; Sivic, 2005], contour matching [Lee, 2009], multiple instance learning (MIL) [Zhu, 2012] and graph mining [Zhang, 2015b] also output object locations, but focus on smaller datasets with only a handful of distinctive object classes. Unsupervised object discovery in large real-world image collections remains challenging due to a high degree of intra-class variation, occlusion, background clutter and the presence of multiple object categories in one image. For this challenging setting, [Cho, 2015] proposes an iterative algorithm which alternates between retrieving image neighbors and localizing salient regions. It also introduces the probabilistic Hough matching algorithm for region similarity score computation which is shown to be effective. Despite promising results, this approach is not formulated as a proper optimization problem and only addresses the discovery of a single object per image. These limitations are addressed in Chapters 2 and 3 with OSD and rOSD respectively. Another line of approaches to unsupervised object discovery focuses on learning object-centric image representations by decomposing images into objects [Burgess, 2019; Engelcke, 2020; Greff, 2019; Locatello, 2020; Monnier, 2021]. These techniques do not scale up (yet) to large natural image collections, and focus mostly on small datasets containing simple shapes in constrained environments. Contrary to them, we focus in this thesis on extracting from natural, in-the-wild images object locations in forms of bounding boxes and identifying pairs of images that contain objects of the same category.

**Image colocalization and object cosegmentation** are closely related to unsupervised object discovery. While the latter does not suppose any prior information on the input images, the former assumes that all input images contain objects of a single category. The goal of image colocalization is to find bounding boxes around these objects. [Tang, 2014] formulates image colocalization as an optimization problem, with inspirations from graph cut and discriminative clustering techniques. [Joulin, 2014] extends this approach to video and solves it with Frank-Wolfe [Frank, 1956] algorithm for efficiency. Observing that supervised object detectors often assign high scores to only a small number of region proposals, [Li, 2016] propose to mimic this behavior by training a classifier to minimize the entropy of the scores it gives to region proposals. [Wei, 2017a; Wei, 2019] localize objects by clustering pixels with high activations in feature maps from CNNs pre-trained in ImageNet. Since there are not many works on unsupervised object discovery for in-the-wild images, we use some of these works as baselines in our experiments, comparing to them in both image colocalization and object discovery tasks. In the former case,

we run our methods without any modification separately on all classes of the tested dataset while in the latter case, we run the colocalization baselines without modification on the entire dataset.

Object cosegmentation aims instead to find masks of the common objects. [Rother, 2006] is the first to consider inter-image consistency to find foreground object segmentation. This work and its immediate successors [Mukherjee, 2009; Hochbaum, 2009] consider pairs of images that figure the same foreground object as input. Subsequent works [Joulin, 2010; Kim, 2011; Rubinstein, 2013; Chen, 2014] extend this setting, and consider multiple images, with possibly some that do not have a common foreground object. These works can be divided into two main groups: graph-based [Lee, 2015; Quan, 2016a; Jerripothula, 2016] and clustering-based [Joulin, 2010; Kim, 2011]. More recent works [Quan, 2016a; Hsu, 2018a; Li, 2019; Hsu, 2019] propose to use pre-trained CNN features for image classification to boost co-segmentation performance. Some of these methods [Quan, 2016a] simply replace hand-crafted features with deep features while others propose end-to-end pipelines for object co-segmentation [Hsu, 2018a; Li, 2019; Hsu, 2019]. Unsupervised object discovery is arguably more challenging than image colocalization and object cosegmentation since it does not impose or exploit any assumptions on the input images.

**Weakly-supervised object localization** is related to UOD but take advantage of image-level labels. It considers scenarios where the input dataset contains image-level labels [Choe, 2020]. Most recent weakly-supervised object localization methods typically localize objects by fine-tuning a pre-trained neural network with the available labels then exploiting object location information in its convolutional features in form of saliency maps [Zhou, 2016; Selvaraju, 2017; Chattopadhay, 2018]. Since CNNs tend to learn category-discriminating features, various strategies for improving the quality of features for localization have been proposed such as adversarial erasing [Zhang, 2018a; Choe, 2019], pseudo supervision [Zhang, 2018b], inter-image consistency [Zhang, 2020e] or leveraging self-supervised pre-training [Baek, 2020]. Similar to these works, some of the methods for unsupervised object discovery proposed in this thesis utilize features pre-trained neural networks but we do not fine-tune them with additional labels.

## 1.3 Active learning strategies for Weakly-Supervised Object Detection

### 1.3.1 Motivation

Unsupervised learning is attractive due to its potential for leveraging an unlimited amount of unlabeled data. However, unsupervised methods often lag far behind supervised ones in terms of performance. In practice, we often have access to a small amount of full annotation or a weak form of annotation. In the case of object detection, full annotation includes a tight bounding box and a class label for each object in the image. While obtaining the bounding box is often laborious, especially when the image contains multiple objects (crowds of people, groups of animals, etc.), the label showing the presence of an object class is much easier to

obtain. Its annotation cost is much lower than the bounding box form (1 second/class *vs.* 35 seconds/box) and it can even be obtained automatically, *e.g.*, leveraging tags on online photos, photo captions in media or time-stamped movie scripts. As a result, weakly-supervised object detection (WSOD), an annotation-efficient alternative to fully-supervised object detection which requires only tags about the presence or absence of object classes in the image, has recently attracted a lot of attention. Several popular methods [Bilen, 2016; Cinbis, 2017; Tang, 2017; Ren, 2020a] formulate weakly-supevised object detection as a multiple instance learning [Foulds, 2010] problem where images are bags and pre-computed region proposals are instances. During training, the model learns to classify bags into correct categories using scores aggregated from the region scores. At inference time, the region scores are used to produce detection results. The performance of weakly-supervised object detectors has improved over the years, especially after the introduction of WSDDN [Bilen, 2016], a neural network-based model, recently reaching an AP50 of 56.8 and 26.4 respectively on VOC2007 and COCO datasets with [Huang, 2020].

Trained with only image tags, which do not contain any information about object position, weakly-supervised object detectors are often confused about object extent and locations. Often, they only find, and quite naturally, the most discriminative parts of objects instead of the entire ones since signals from these parts are enough to correctly classify the images in the multiple instance learner. They also tend to detect groups of objects instead of individual ones when many of them are close in the images. Finally, they struggle to detect all objects in the images as finding one of them is sufficient to solve the classification task. Many efforts have attempted to remedy these issues with better pseudo labels [Tang, 2018a; Ren, 2020b], better region representation [Ren, 2020b; Huang, 2020] and better optimization [Arun, 2019; Wan, 2019], and they have led to some improvements. However, such confusions remain and the performance of weakly-supervised object detectors is still far behind their fully-supervised counterparts. For example, a Fast-RCNN with the same backbone obtains an AP50 of 66.9 and 38.6 respectively on VOC2007 and COCO datasets. Some recent works [Pan, 2019; Biffi, 2020] propose to narrow this gap and achieve a better trade-off between annotation cost and detection performance by annotating a randomly selected set of training images and train with a mixed of weak and full supervision. We believe that better selection strategies than random can achieve an even better trade-off and propose to consider active learning strategies for image selection.

### 1.3.2 Contributions

In Chapter 6, we proposes to fine-tune a trained weakly-supervised object detector with annotated samples selected using active learning techniques [Geifman, 2017; Sener, 2018; Brust, 2019; Choi, 2021]. By carefully choosing images that are most relevant to improving the current model, we aim to improve the model performance as much as possible while using as little additional annotation as possible, achieving a better trade-off between annotation cost and detection performance than both weakly- and fully-supervised object detection. We begin with a training set where images have only class annotation and train a detector in a weakly-supervised manner. Then, we run multiple cycles in which an active learning strategy is used to select images from the training set that do not have yet bounding-box labels and annotate them with

a bounding box around each object in the images. After the selection, the weakly-supervised model is fine-tuned with all the images in the dataset, using bounding-box annotations when they are available and image tags otherwise.

We introduce BiB [Vo, 2022], an active learning strategy tailored for this pipeline. It targets the known modes of confusion of weakly-supervised object detectors: Objects *vs.* object parts and objects *vs.* groups of objects. In particular, it uses the presence of BiB pairs – pairs of predictions in the same image such that one is *contained* in the other – as an indicator of model confusion, and selects a diverse set of images amongst those on which the model is confused. We show that BiB significantly outperforms all other active learning strategies proposed so far in the setting. More importantly, it boosts the detection performance of weakly-supervised object detectors significantly, reaching 97% of the performance of fully-supervised Fast RCNN [Girshick, 2015] with only 10% of fully-annotated images on VOC07. On COCO, using on average 10 fully-annotated images per class, that is about 1% of training images fully-annotated, BiB also cuts the performance gap (in AP) between weakly-supervised and fully-supervised detectors by over 70%, showing a good trade-off between performance and data efficiency.

### 1.3.3 Related Work

**Weakly-supervised object detection** is an annotation-efficient alternative to fully-supervised object detection which only requires image-level labels (object categories) for training a detector. It is typically formulated as a multiple instance learning problem [Dietterich, 1997], where images are bags and region proposals [Uijlings, 2013; Zitnick, 2014] are instances. The model is trained to classify images using scores aggregated from their region proposals. Through this process, it also learns to distinguish *object* from *non-object* regions. Since training involves solving a non-convex optimization problem, adapted initialization and regularization techniques [Deselaers, 2010; Kumar, 2010; Song, 2014a; Song, 2014b; Cinbis, 2017] are necessary for good performance. [Bilen, 2016] proposes WSDDN, a CNN-based model for WSOD, which is further improved in subsequent works [Diba, 2017; Jie, 2017a; Tang, 2017; Tang, 2018a; Ren, 2020a]. [Tang, 2017] proposes OICR which refines WSDDN's output with parallel detector heads in a self-training fashion. Trained with only image-level labels, weakly-supervised object detectors are often confused between object parts and objects, or between objects and groups of objects [Ren, 2020a]. Although recent mitigating efforts with better pseudo labels [Tang, 2018a; Ren, 2020a], better representations [Ren, 2020a; Huang, 2020] or better optimization [Arun, 2019; Wan, 2019] have achieved some successes, the confusion issues of weakly-supervised detectors remain due to the lack of an operational definition of objects and their performance is still far behind that of fully-supervised counterparts. In this work, we show that fine-tuning weakly-supervised detectors with strong annotation on *a few carefully selected* images can alleviate these limitations and significantly narrow the gap between weakly- and fully-supervised object detectors.

**Semi-supervised object detection** methods exploits a mix of a few fully-annotated and many unlabelled data. Two dominant strategies have arisen among these methods using consis-

tency [Jeong, 2019; Tang, 2021b] and pseudo-labeling [Radosavovic, 2018; Wang, 2018a; Zoph, 2020; Li, 2020; Sohn, 2020b; Xu, 2021]. The latter can be further extended with strategies inspired by active learning [Wang, 2018a; Li, 2020] for selecting boxes to be annotated by people.

**Combining weakly- and semi-supervised object detection.** These approaches seek a better trade-off between performance and annotation cost than individual strategies. All images from the training set have weak labels and a subset is also annotated with bounding boxes. This setup enables the exploration of the utility of additional types of weak labels, e.g., points [Ren, 2020b; Chen, 2021] or scribbles [Ren, 2020b]. Others leverage fully-annotated images to train detectors that can correct wrong predictions of weakly-supervised detectors [Pan, 2019] or compute more reliable pseudo-boxes [Biffi, 2020]. Similarly to [Pan, 2019; Biffi, 2020], we train a detector with only a few annotated images, but contrary to them, we focus on how to best select the images to annotate so as to maximize the performance of the detector.

**Active learning for object detection** aims at carefully *selecting* images to be fully annotated, in order to minimize human annotation efforts. Most methods exploit *data diversity* [Geifman, 2017; Sener, 2018] or *model uncertainty* [Brust, 2019; Choi, 2021] to identify such images. These strategies, originally designed for generic classification tasks [Settles, 2009], have been recently adapted to object detection [Yuan, 2021b; Choi, 2021], a complex task involving both classification (object category) and regression (bounding box). Data diversity can be ensured by selecting data samples using image features and applying k-means [Zhdanov, 2019], k-means++ initialization [Haussmann, 2020] or identifying a core-set – a *representative* subset of a dataset [Geifman, 2017; Sener, 2018; Agarwal, 2020]. Model uncertainty for active learning can be computed from image-level scores aggregated from class predictions over boxes [Brust, 2019; Haussmann, 2020; Pardo, 2021], comparing predictions of the same image from its different corrupted versions [Kao, 2018; M, 2020; Elezi, 2021] or from different steps of model training [Roy, 2018; Huang, 2021], voting over predictions from an ensemble of networks [Beluch, 2018; Chitta, 2019; Haussmann, 2020], using Bayesian Neural Networks [Gal, 2017; Haussmann, 2020] or single forward networks mimicking an ensemble [Choi, 2021; Yuan, 2021b]. Multiple other strategies have been proposed for selecting informative, difficult or confusing samples to annotate by learning to discriminate between labeled and unlabeled data [Gissin, 2019; Ebrahimi, 2019; Ebrahimi, 2020; Zhang, 2020a], learning to predict the detection loss [Yoo, 2019], the gradients [Ash, 2020] or the influence of data on gradient [Liu, 2021c]. In contrast to classical active learning methods in which the initial model is trained in a fully-supervised fashion using a randomly sampled initial set of images, our initial model is only trained with weakly-annotated data. This is a challenging setting, but often encountered in practice when new collections of data arrive only with weak annotations and significant effort is required to select which images to annotate manually prior to active learning.

**Combining weak supervision and active learning.** Closer to us, [Desai, 2019; Fang, 2020; Pardo, 2021] investigate how weakly-supervised learning and active learning can be conducted

together in the context of object detection. [Desai, 2019] proposes to use clicks in the center of the object as weak labels which include localization information and are stronger than image-level tags. [Pardo, 2021] also mixes strong supervision, tags and pseudo-labels in an active learning scenario. Both [Desai, 2019; Pardo, 2021] rely on Faster R-CNN [Ren, 2015a] and [Fang, 2020] on FPN [Lin, 2017] – detectors that are hard to train only with weak labels. All start with 10% of the dataset fully labeled, which is more than the total amount of fully annotated data we ever consider in this work.

# Part I

# Unsupervised Object Discovery

# Chapter 2

# Unsupervised Image Matching and Object Discovery as Optimization

Objectives

Supervised machine learning is a powerful framework but it relies on ever-growing human annotation efforts. As a way to mitigate this serious problem, as well as to serve specific applications, unsupervised learning has emerged as an important field of research. In computer vision, unsupervised learning comes in various guises. We focus here on the unsupervised discovery and matching of object categories among images in a collection, following the seminal work of [Cho, 2015]. We show that the original approach can be reformulated and solved as a proper optimization problem. Experiments on several benchmarks establish the merit of our approach.
This work, done in collaboration with Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez and Jean Ponce, has appeared in Proceedings of the IEEE/CVF Conference in Computer Vision and Pattern Recognition (CVPR) 2019.

**Contents**


2.1 Introduction . . . **22**
2.2 Proposed Approach . . . **23**
2.2.1 Problem Statement . . . 23
2.2.2 Relaxing the Problem . . . 24
2.2.3 Solving the Dual Problem . . . 25
2.2.4 Solving the Primal Problem . . . 25
2.2.5 Rounding the Solution and Greedy Ascent . . . 26
2.2.6 Ensemble Post Processing . . . 26
2.3 Similarity Model . . . **28**
2.3.1 Confidence Score . . . 28
2.3.2 Stand-out Score . . . 29
2.4 Experiments and Results . . . **30**
2.5 Conclusion and Limitations . . . **36**

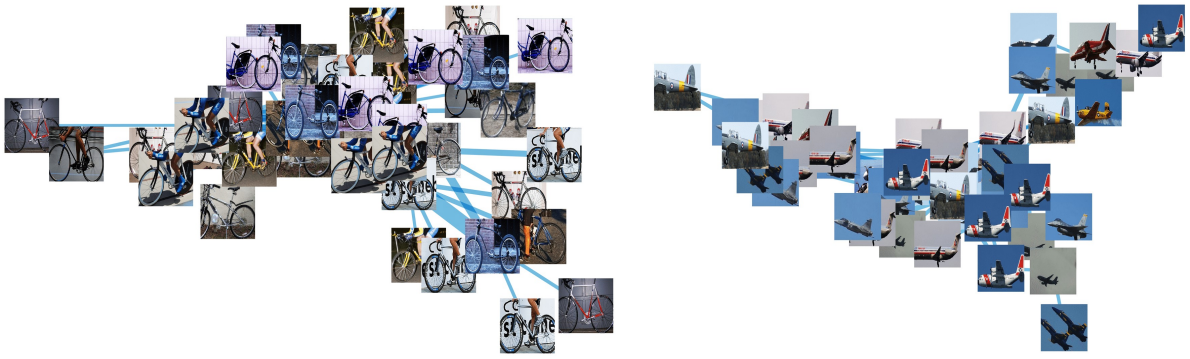


Figure 2.1 – The proposed optimization-based method automatically discovers links between images that depict similar objects. This figure shows two image clusters that emerge as a by-product of this approach on the VOC_6x2 object recognition dataset that mixes 6 classes under two viewpoints. See Section 2.4 for details.

## 2.1 Introduction

We introduce in this chapter an optimization formulation for unsupervised object discovery which leverages the implicit graph structure of image sets. Any collection of images, say, those found on the Internet, or more modestly, in a dataset such as Pascal VOC2007 [Everingham, 2007], admits a natural graph representation, where nodes are the pictures themselves, and edges link pairs of images with similar visual content. In *supervised* image categorization task [Lazebnik, 2006; Krizhevsky, 2012; Simonyan, 2015a; He, 2016] or object detection [Felzenszwalb, 2010; Ren, 2015a; He, 2017], both the graph structure and the visual content are clearly defined: Annotators typically sort the images into "bags", each one intended to represent some "object", "scene" or, say, "action" class ("horse", "forest", "playing tennis", etc.). Two nodes are linked by an edge when they are associated with the same bag, and each class is empirically defined by the images (or some manually-defined axis-aligned rectangular regions within) in the corresponding connected component of the graph. In *weakly-supervised* cosegmentation [Joulin, 2010; Kim, 2011; Rubinstein, 2013] or colocalization [Deselaers, 2010; Joulin, 2014; Tang, 2014] tasks, on the other hand, the graph is fully connected, and all images are supposed to contain instances of a (few) object categories, say, "horse", "grass", "sky", "background". Manual intervention is reduced to selecting which images to put into each bag, and the visual content, in the form of regions defined by pixel-level symbolic labels or bounding boxes associated with one of the predefined categories, is discovered using a clustering algorithm. [1]

We address the much more difficult problem of *fully-unsupervised* image matching and object discovery, where both the graph structure and a model of visual content in the form of object bounding boxes must be extracted from the native data without any manual intervention. We

1. In both the cases of supervised image categorization/object detection and weakly-supervised cosegmentation/colocalization, once the graph structure and the visual content have been identified at *training time*, these can be used to learn a model of the different object classes and add nodes, edges, and possibly additional bounding boxes at *test time*.

build directly on the work of [Cho, 2015] (see [Kwak, 2015] for related work): Given an image and its neighbors, assumed to contain the same object, a robust matching technique exploits both appearance and geometric consistency constraints to assign confidence and saliency ("stand-out") scores to region proposals in this image. The overall discovery algorithm alternates between *localization* steps where the neighbors are fixed and the regions with top saliency scores are selected as potential objects, and *retrieval* steps where the confidence of the regions within potential objects are used to find the nearest neighbors of each image. After a fixed number of steps, the region with top saliency in each image is declared to be the object it contains ([Cho, 2015] focuses on discovering the single most prominent object of each image). Empirically, this method has been shown to give good results. However, it does not formulate image matching and object discovery as a proper optimization problem, and there is no guarantee that successive iterations will improve some objective measure of performance. The aim of this chapter is to remedy this situation.

## 2.2 Proposed Approach

### 2.2.1 Problem Statement

Let us consider a set of $n$ images, where image $i$ contains $p_i$ rectangular region proposals, with $i$ in $\{1 \dots n\}$. We assume that the images are equipped with some implicit graph structure, where there is a link between two images when the second image contains at least one object from a category depicted in the first one, and our aim is to discover this structure, that is, find the links and the corresponding objects. To this end, we propose in the following the object and structure discovery, or OSD, formulation. Let us define an indicator variable $x_i^k$, whose value is 1 when region number $k$ of image $i$, itself denoted as $(i, k)$, corresponds to a "foreground object" (visible in large part and from a category that occurs multiple times in the image collection), and 0 otherwise. We collect all the variables $x_i^k$ associated with image $i$ into an element $x_i$ of $\{0,1\}^{p_i}$, and concatenate all the variables $x_i$ into an element $x$ of $\{0,1\}^{\sum_{i=1}^{n} p_i}$. Likewise, let us define an indicator variable $e_{ij}$, whose value is 1 if image $j$ contains an object also occurring in image $i$, with $1 \leq i, j \leq n$ and $j \neq i$, and 0 otherwise, collect all the variables $e_{ij}$ associated with image $i$ into an element $e_i$ of $\{0,1\}^n$, and concatenate all the variables $e_i$ into an $n \times n$ matrix $e$ with rows $e_i^T$. Note that we can use $e$ to define a neighborhood for each image in the set: Image $j$ is a neighbor of the image $i$ iff $e_{ij} = 1$. By definition, $e$ defines an undirected graph if $e$ is symmetric and a directed one otherwise. Let us also denote by $S_{ij}^{kl}$ the similarity between regions $k$ and $l$ of images $i$ and $j$, which represents the likelihood that the two regions correspond to objects of the same category, and by $S_{ij}$ the $p_i \times p_j$ matrix with entries $S_{ij}^{kl}$.

We propose to maximize with respect to $x$ and $e$ the objective function

$$S(x,e) = \sum_{\substack{i,j=1 \\ j \neq i}}^{n} e_{ij} \sum_{\substack{1 \leq k \leq p_i \\ 1 \leq l \leq p_j}} S_{ij}^{kl} x_i^k x_j^l = \sum_{\substack{i,j=1 \\ j \neq i}}^{n} x_i^T [e_{ij} S_{ij}] x_j. \tag{2.1}$$

Intuitively, maximizing $S(x, e)$ encourages building edges between images $i$ and $j$ that contain

regions $k$ and $l$ with a strong similarity $S_{ij}^{kl}$. Of course we would like to impose certain constraints on the $x$ and $e$ variables. The following cardinality constraints are rather natural:

- An image should not contain more than a prededined number of objects, say $\nu$,

$$\forall\ i \in \{1 \dots n\},\ \sum_k x_{ik} \leq \nu. \tag{2.2}$$

- An image should not match more than a predefined number of other images, say $\tau$,

$$\forall\ i \in \{1 \dots n\},\ \sum_j e_{ij} \leq \tau. \tag{2.3}$$

**Assumptions.** We will suppose from now on that $S_{ij}$ is elementwise nonnegative, but not necessarily symmetric (the similarity model we explore in Section 3 is asymmetrical). Likewise, we will assume that the binary matrix $e$ has a zero diagonal but is not necessarily symmetric.

Under these assumptions, $S$ is a supermodular cubic pseudo-Boolean function [Boros, 2002]. Without constraints, this type of functions can be maximized in polynomial time using a max-flow algorithm [Billionnet, 1985] (in the case of $S(x, e)$, which does not involve linear and quadratic terms, the solution is of course trivial without constraints, and amounts to setting all $x_i^k$ and $e_{ij}$ with $i \neq j$ to 1). When the cardinality constraints (2.2-2.3) are added, this is not the case anymore, and we have to resort to a gradient ascent algorithm as explained next.

### 2.2.2 Relaxing the Problem

Let us first note that, for binary variables $x_i^k$, $x_j^l$ and $e_{ij}$, $S(x, e)$ can be equivalently rewritten as

$$S(x, e) = \sum_{\substack{i,j=1 \\ j \neq i}}^{n} \sum_{\substack{1 \leq k \leq p_i \\ 1 \leq l \leq p_j}} S_{ij}^{kl} \min(e_{ij}, x_i^k, x_j^l), \tag{2.4}$$

with $S_{ij}^{kl} \geq 0$. Relaxing our problem so all variables are allowed to take values in $[0, 1]$, our objective becomes a sum of concave functions, and thus is itself a concave function, defined over the convex set (hyperrectangle) $[0, 1]^N$, where $N$ is the total number of variables. This is the standard tight concave continuous relaxation of supermodular functions.

The Lagrangian associated with our relaxed problem is

$$K(x, e; \lambda, \mu) = S(x, e) - \sum_{i=1}^{n} [\lambda_i (x_i \cdot \mathbb{1}_{p_i} - \nu) + \mu_i (e_i \cdot \mathbb{1}_n - \tau)], \tag{2.5}$$

where $\lambda = (\lambda_1, \dots, \lambda_n)^T$ and $\mu = (\mu_1, \dots, \mu_n)^T$ are *positive* Lagrange multipliers. The function $S(x, e)$ is concave and the primal problem is strictly feasible; hence Slater's conditions [Slater, 1950] hold, and we have the following equivalent primal and dual versions of our problem

$$\begin{cases} \max_{(x,e) \in D} \inf_{\lambda, \mu \geq 0} K(x, e; \lambda, \mu), \\ \min_{\lambda, \mu \geq 0} \sup_{(x,e) \in D} K(x, e; \lambda, \mu), \end{cases} \tag{2.6}$$

where the domain $D$ is the Cartesian product of $[0,1]^{\sum_i p_i}$ and the space of $n \times n$ matrices with entries in $[0,1]$ and a zero diagonal. With slight abuse we denote it $D = [0,1]^N$, with $N = \sum_i p_i + n(n-1)$.

### 2.2.3 Solving the Dual Problem

We propose to solve the dual problem with a subgradient descent approach. Starting from some initial values for $\lambda^0$ and $\mu^0$, we use the update rule

$$\begin{cases} \lambda_i^{t+1} = [\lambda_i^t + \alpha(x_i^t \cdot \mathbb{1}_{p_i} - \nu)]_+, \\ \mu_i^{t+1} = [\mu_i^t + \beta(e_i^t \cdot \mathbb{1}_n - \tau)]_+, \end{cases} \tag{2.7}$$

where $[x]_+$ denotes the positive part of a scalar $x$, $k \geq 0$, $\alpha$ and $\beta$ are fixed step sizes, $x_i^t \cdot \mathbb{1}_{p_i} - \nu$ and $e_i^t \cdot \mathbb{1}_n - \tau$ are respectively the negative of the subgradients of the Lagrangian with respect to $\lambda_i$ and $\mu_i$ in $\lambda_i^t$ and $\mu_i^t$, and

$$(x^t, e^t) \in \operatorname{argmax}_{(x,e)\in[0,1]^N} K(x, e; \lambda^t, \mu^t). \tag{2.8}$$

As shown in Appendix A, for fixed values of $\lambda$ and $\mu$, our Lagrangian is a *supermodular* pseudo-Boolean function of binary variables sets $x$ and $e$. This allows us to take advantage of the following direct corollary of [Bach, 2013, Prop. 3.7].

**Proposition 2.2.1.** *Let $f$ denote some supermodular pseudo-Boolean function of $n$ variables. We have*

$$\max_{x\in\{0,1\}^n} f(x) = \max_{x\in[0,1]^n} f(x), \tag{2.9}$$

*and the set of maximizers of $f(x)$ in $[0,1]^n$ is the convex hull of the set of maximizers of $f$ on $\{0,1\}^n$.*

In particular, we can take

$$(x^t, e^t) \in \operatorname{argmax}_{(x,e)\in\{0,1\}^N} K(x, e; \lambda^t, \mu^t). \tag{2.10}$$

As shown in [Billionnet, 1985; Boros, 2002], the corresponding supermodular cubic pseudo-Boolean function optimization problem is equivalent to a maximum stable set problem in a bipartite *conflict graph*, which can itself be reduced to a maximum-flow problem. See Appendix A for details.

Note that the size of the min-cut/max-flow problems that have to be solved is conditioned by the number of nonzero $S_{ij}^{kl}$ entries, which is upper-bounded by $n^2p^2$ when the matrices $S_{ij}$ are dense (denoting $p = \max\{p_i\}$). This is prohibitively high given that, in practice, $p$ is between 1000 and 4000. To make the computations manageable, we set all but between 100 and 1000 (depending on the dataset's size) of the largest entries in $S_{ij}$ to zero in our implementation.

### 2.2.4 Solving the Primal Problem

Once the dual problem is solved, as argued by [Nedić, 2009] and [Bach, 2013], an approximate solution of the primal problem can be found as a running average of the primal sequence $(x^t, e^t)$ generated as a by-product of the sub-gradient method:

$$\hat{x} = \frac{1}{T}\sum_{t=0}^{T-1} x^t, \quad \hat{e} = \frac{1}{T}\sum_{t=0}^{T-1} e^t \tag{2.11}$$

after some number $T$ of iterations. Note the scalars $\hat{x}_i^k$ and $\hat{e}_{ij}$ lie in $[0, 1]$ but do not necessarily verify the constraints (2.2) and (2.3). Theoretical guarantees on these values can be found under additional assumptions in [Nedić, 2009; Bach, 2013].

### 2.2.5 Rounding the Solution and Greedy Ascent

Two problems remain to be solved: The solution $(\hat{x}, \hat{e})$ found now belongs to $[0, 1]^N$ instead of $\{0, 1\}^N$, and it may not satisfy the original constraints. Note, however, that given some $i$ in $\{1, \dots, n\}$ and fixed values for $e$ and all $x_j$ with $j \neq i$, the function $S$ can be rewritten as

$$S(x, e) = \sum_{j \neq i} (e_{ij} S_{ij} + e_{ji} S_{ji}^T) x_j + C, \tag{2.12}$$

where $C$ is a term that does not depend on $x_i$. The maximum value of $S$ in this case, given the constraints, is therefore obtained by setting to 1 exactly the $\nu$ entries of $x_i$ corresponding to the $\nu$ largest entries of the vector $\sum_{j \neq i} (e_{ij} S_{ij} + e_{ji} S_{ji}^T) x_j$. Likewise, given some $i$ in $\{1, \dots, n\}$ and a fixed value of $x$, $S$ is rewritten as

$$S(x, e) = \sum_{j \neq i} x_i^T S_{ij} x_j + C, \tag{2.13}$$

with $C$ is $e_i$-independent. The maximum value of $S$ is thus reached by setting to 1 exactly the $\tau$ entries of $e_i$ corresponding to the $\tau$ largest scalars $x_i^T S_{ij} x_j$ for $j \neq i$ in $\{1 \dots n\}$. This suggests the following approach to rounding up the solution, where the variables $x_i$ are updated sequentially in an order specified by some random permutation $\sigma$ of $\{1, \dots, n\}$, before the variables $e_i$ are updated in parallel. Given the permutation $\sigma$, Algorithm 2.1 turns the running average $(\hat{x}, \hat{e})$ of the primal sequence into a discrete solution $(x, e)$ that satisfies the conditions (2.2) and (2.3).

Note that there is no preferred order for the image indices. This actually suggests repeating this procedure with different random permutations until the variables $x$ and $e$ do not change anymore or some limit on the number of iterations is reached. This iterative procedure can be seen as a greedy ascent procedure over the discrete variables of interest. Note that by construction the terms in the left and right sides of Equations (2.2) and (2.3) are equal at the optimum.

**Algorithm 2.1:** Greedy block coordinate ascent algorithm.

**Input:** Number of images $n$, score matrices $(S_{ij})_{1\leq i,j\leq n}$, parameters $\nu$ and $\tau$, running average $(\hat{x}, \hat{e})$ from the continuous optimization.

**Result:** Discrete, feasible solution $(x, e)$.

Initialize $x = \hat{x}$, $e = \hat{e}$.
$\sigma \leftarrow$ `rand-perm`$([1..n])$ ▷ Generate a random permutation of $[1..n]$
**for** $i = 1$ to $n$ **do**
  $C_x \leftarrow \sum_{j\neq\sigma(i)}^{n}(e_{\sigma(i)j}S_{\sigma(i)j} + e_{j\sigma(i)}S^T_{j\sigma(i)})x_j$ ▷ Comp. coef. in $S(x, e)$ of elem. of $x_{\sigma(i)}$
  $k_1, \ldots, k_\nu \leftarrow$ `find-max-indices`$(C_x, \nu)$ ▷ Find ids of $\nu$ largest elem. in $C_x$
  $x_{\sigma(i)} \leftarrow 0$ ▷ Re-initialize $x_{\sigma(i)}$
  **for** $t = 1$ to $\nu$ **do**
    $x^{k_t}_{\sigma(i)} \leftarrow 1$ ▷ Assign elements with indices $k_1, \ldots, k_\nu$ to 1
  **end**
**end**
**for** $i = 1$ to $n$ **do**
  $C_e \leftarrow [x_i^T S_{i1}x_1; x_i^T S_{i2}x_2; \ldots; x_i^T S_{in}x_n]$ ▷ Comp. coef. in $S(x, e)$ of elem. in $e_i$
  $j_1, \ldots, j_\tau \leftarrow$ `find-max-indices`$(C_e, \tau)$ ▷ Find ids of $\tau$ largest elem. in $C_e$
  $e_i \leftarrow 0$ ▷ Re-initialize $e_i$
  **for** $t = 1$ to $\tau$ **do**
    $e_{ij_t} \leftarrow 1$ ▷ Assign elements with indices $j_1, \ldots, j_\tau$ to 1
  **end**
**end**
Return $(x, e)$.

### 2.2.6 Ensemble Post Processing

The parameter $\nu$ can be seen from two different viewpoints: (1) as the maximum number of objects that may be depicted in an image, or (2) as an upper bound on the total number of object region *candidates* that are under consideration in a picture. Both viewpoints are equally valid but, following [Cho, 2015], we focus in the rest of this chapter on the second one, and present in this section a simple heuristic for selecting one final object region among these candidates. Concretely, since using random permutations during greedy ascent provides a different solution for each run of our method, we propose to apply an *ensemble method* to stabilize the results and boost performance in this selection process, itself viewed as a post-processing stage separate from the optimization part.

Let us suppose that after $L$ independent executions of the greedy ascent step, we obtain $L$ solutions $(x(l), e(l))$, $1 \leq l \leq L$. We start by combining these solutions into a single discrete pair $(\bar{x}, \bar{e})$ where $\bar{x}$ and $\bar{e}$ satisfy $\bar{x}_i^k = \max_{1\leq l\leq L} x_i^k(l)$ and $\bar{e}_{ij} = \max_{1\leq l\leq L} e_{ij}(l)$. This way of combining the individual solutions can be seen as a *max pooling* procedure. We have also tried *average pooling* but found it less effective. Note that after this intermediate step, an image might violate any of the two constraints (2.2-2.3). This is not a problem in this postprocessing stage of our method. Indeed, we next show how to use $\bar{x}$ and $\bar{e}$ to select a *single* object proposal for each image.

We choose a single proposal for each image out of those retained in $\bar{x}$ (proposals $(i, k)$ s.t.

$\bar{x}_i^k = 1$). To this end, we rank the proposals in image $i$ according to a score $u_i^k$ defined for each proposal $(i,k)$ as

$$u_i^k = \bar{x}_i^k \sum_{j \in \mathcal{N}(i,k)} \max_{l|\bar{x}_j^l=1} S_{ij}^{kl}, \tag{2.14}$$

where $\mathcal{N}(i,k)$ is composed of the $\tau$ images represented by the 1 entries in $\bar{e}_i$ that have the largest similarity to $(i,k)$ as measured by $\max_{l|\bar{x}_j^l=1} S_{ij}^{kl}$. Finally, we choose the proposal in image $i$ with maximum score $u_i^k$ as the final object region. Note that the graph of images corresponding to these final object regions can be retrieved by computing $e$ that maximizes the objective function given the value of $x$ defined by these regions as in the greedy ascent. Also, the method above can be generalized to selecting more than one proposal per image using the defined ranking but following [Cho, 2015], we focus in this chapter on finding only the most prominent object in each image.

## 2.3 Similarity Model

Let us now get back to the definition of the similarity score $S_{ij}$. As advocated by [Cho, 2015], a rectangular region which is a tight fit for a compact object (the *foreground*) should better model this object than a larger region, since it contains less *background*, or than a smaller region (a *part*) since it contains more foreground. [Cho, 2015] only implement the first constraint, in the form of a *stand-out* score. We discuss in this section how to implement these ideas in the optimization context of this work.

### 2.3.1 Confidence Score

Following [Cho, 2015], the confidence score between proposal $k$ of image $i$ and proposal $l$ of image $j$ can be defined as

$$s_{ij}^{kl} = a_{ij}^{kl} \sum_{o \in O} g(r_i^k, r_j^l, o) \sum_{\substack{1 \le k' \le p_i \\ 1 \le l' \le p_j}} g(r_i^{k'}, r_j^{l'}, o) a_{ij}^{k'l'}, \tag{2.15}$$

where $a_{ij}^{kl}$ is a similarity term based on appearance alone, using the WHO (whiten HOG) descriptor [Dalal, 2005; Hariharan, 2012] in our case; $r_i^k$ and $r_j^l$ denote the image rectangles associated with the two proposals; $o$ is a discretized offset (translation plus a scale factor) taking values in $O$; and $g(r,s,o)$ measures the geometric compatibility between $o$ and the rectangles $r$ and $s$. Intuitively, $s_{ij}^{kl}$ scales the appearance-only score $a_{ij}^{kl}$ by a geometric-consistency term akin to a generalized Hough transform [Ballard, 1981], see [Cho, 2015] for details.

Note that we can rewrite Equation (2.15) as

$$s_{ij}^{kl} = b_{ij}^{kl} \cdot c_{ij}, \tag{2.16}$$

where $b_{ij}^{kl}$ is the vector of dimension $|O|$ with entries $a_{ij}^{kl} g(r_i^k, r_j^l, o)$, and $c_{ij} = \sum_{k',l'=1}^{p} b_{ij}^{k'l'}$. The $p_i p_j$ vectors $b_{ij}^{kl}$ and the vector $c_{ij}$ can be precomputed with time and storage cost of $\mathcal{O}(p^2|O|)$.

Each term $s_{ij}^{kl}$ can then be computed in $\mathcal{O}(|O|)$ time, and the matrix $S_{ij}$ can thus be computed with a total time and space complexity of $\mathcal{O}(p^2|O|)$.

The score $s_{ij}^{kl}$ defined by Equation (2.15) depends on the number of region proposals per images, which may introduce a bias for edges between images that contain many region proposals. It is thus desirable to *normalize* this score by defining it instead as

$$s_{ij}^{kl} = \frac{1}{p_i p_j} b_{ij}^{kl} \cdot c_{ij}. \tag{2.17}$$

### 2.3.2 Stand-out Score

Let us define $P_i^k$ as the set of regions in image $i$ that are *parts* of region $(i,k)$, i.e., a large percentage of their area is included in region $(i,k)$. Let us also define $B_i^k$ as the set of regions in image $i$ that form the *background* for $(i,k)$, i.e., a large part of the latter is included in these regions. With this definition, we consider $(i,k') - (j,l')$ a background match of $(i,k) - (j,l)$ if $(i,k') \in B_i^k$ and $(j,l') \in B_j^l$. Let $r_i^k$ denote the actual rectangular image region associated with $(i,k)$, and let $A(r)$ denote the area of some rectangle $r$. We define $P_i^k$ as

$$P_i^k = \{l \,:\, A(r_i^k \cap r_i^l) > \rho A(r_i^l)\}, \tag{2.18}$$

for some suitable value of $\rho$, e.g., 0.5. Likewise, $B_i^k$ is defined as

$$B_i^k = \{l : A(r_i^k \cap r_i^l) > \delta A(r_i^k) \text{ and } A(r_i^l) > \gamma A(r_i^k)\}, \tag{2.19}$$

for suitable values of $\delta$ and $\gamma$, e.g., 0.8 and 2. Following [Cho, 2015], we define the *stand-out score* of a match $(i,k) - (j,l)$ as the difference in confidence score to its most confident background match

$$S_{ij}^{kl} = s_{ij}^{kl} - v_{ij}^{kl}, \text{ where } v_{ij}^{kl} = \max_{(k',l') \in B_i^k \times B_j^l} s_{ij}^{k'l'}. \tag{2.20}$$

With this definition, $S_{ij}^{kl}$ may be negative. In our implementation, we threshold these scores so that they are non-negative.

When $B_i^k$ and $B_j^l$ are large, which is generally the case when the regions $r_i^k$ and $r_j^l$ are small, a brute-force computation of $v_{ij}^{kl}$ may be very slow. We propose below instead a simple heuristic that greatly speeds up calculations. Let $Q_{ij}$ denote the set formed by the $q$ matches $(i,k) - (j,l)$ with highest scores $s_{ij}^{kl}$, sorted in increasing order, which can be computed in $\mathcal{O}(p^2 \log p)$ with QuickSort [Hoare, 1961] or $\mathcal{O}(p^2 + q \log q)$ with Median of Medians [Blum, 1973] then QuickSort. The stand-out scores can be computed efficiently by Algorithm 2.2.

The idea is that relatively few high-confidence matches $(i,k') - (j,l')$ in $Q_{ij}$ can be used to efficiently compute many stand-out scores. There is a trade-off between the cost of this step, $\mathcal{O}(\sum_{(k',l') \in Q_{ij}} |P_i^{k'}| |P_j^{l'}|)$, and the number of variables $v_{ij}^{kl}$ it assigns a new value to, $\mathcal{O}(|\cup_{(k',l') \in Q_{ij}} P_i^{k'} \times P_j^{l'}|)$. In practice, we have found that taking $q = 10,000$ is a good compromise, with only about 5% of the stand-out scores being computed in a brute-force manner, and a significant speed-up factor of over 10.

**Algorithm 2.2:** Standout score computation.

**Input:** Top $q$ confident matches $Q_{ij}$, confidence score $s_{ij}$.
**Result:** Stand-out score matrix $S_{ij}$ for all matches between regions in images $i$ and $j$.

Initialize all $v_{ij}^{kl}$ to 0
`// Quickly compute the most confident background matches for most matches`
**foreach** match $(i,k') - (j,l')$ in $Q_{ij}$ **do**
  **foreach** match $(i,k) - (j,l)$ in $P_i^{k'} \times P_j^{l'}$ **do**
    Assign $v_{ij}^{kl} = s_{ij}^{k'l'}$
  **end**
**end**
`// Compute the most confident background matches for the remaining matches`
**for** $k = 1$ to $p_i$ and $l = 1$ to $p_j$ **do**
  **if** $s_{ij}^{kl} > 0$ and $v_{ij}^{kl} = 0$ **then**
    $v_{ij}^{kl} = \max_{(k',l') \in B_i^k \times B_j^l} s_{ij}^{k'l'}$
  **end**
**end**
`// Compute the stand-out score`
**for** $k = 1$ to $p_i$ and $l = 1$ to $p_j$ **do**
  $S_{ij}^{kl} = s_{ij}^{kl} - v_{ij}^{kl}$
**end**

## 2.4 Experiments and Results

**Datasets, proposals and metric.** For our experiments we use the same datasets (Object-Discovery [OD], VOC_6x2 and VOC_all) and region proposals (obtained by the randomized Prim algorithm [RP] [Manen, 2013]) as [Cho, 2015]. OD consists of pictures of three object classes (*airplane*, *horse* and *car*) with outliers not containing any object instance. There are 100 images per category, with 18, 7 and 11 outliers respectively. VOC_all is a subset of the PASCAL VOC2007 train+val dataset obtained by eliminating all images containing only objects marked as *difficult* or *truncated*. *Difficult* and *truncated* objects in remaining images are also discarded. In total, it has 3550 images containing 6661 objects. Finally, VOC_6x2 is a subset of VOC_all containing only 463 images of 6 classes – *aeroplane*, *bicycle*, *boat*, *bus*, *horse* – and *motorbike* from two different views, *left* and *right*.

For evaluation, we use the standard *CorLoc* measure, the percentage of images correctly localized. It is a proxy metric in the case of unsupervised object discovery. An image is "correctly localized" when the intersection over union ($IoU$) between one of the ground-truth regions and the predicted one is at least 0.5. Following [Cho, 2015], we evaluate our algorithm in "separate" and "mixed" settings, which respectively correspond to the colocalization and the true discovery settings. In the former case, the class-wise performance is averaged over classes. In the latter, a single performance is computed over all classes jointly. In our experiments, we use $\nu = 5$, $\tau = 10$ and standout matrices with 1000 non-zero entries unless mentioned otherwise.

**Separate setting.** We firstly evaluate different configurations of our algorithm on the two smaller datasets, OD and VOC_6x2. The performance is governed by three design choices: (1) Using the normalized stand-out score (*NS)* or its unnormalized version, (2) using continuous optimization (*CO*) or variables $x$ and $e$ with all entries equal to one to initialize the greedy ascent procedure, and (3) using the ensemble method (*EM*) or not. In total, we thus have eight configurations to test.

| Method | | | OD | VOC_6x2 |
|---|---|---|---|---|
| [Cho, 2015] | | | 84.2 | 67.7 |
| [Cho, 2015] (our execution) | | | 84.2 | 67.6 |
| w/o EM | w/o CO | w/o NS | 81.9 ± 0.9 | 65.9 ± 1.0 |
| | | w NS | 83.1 ± 0.8 | 67.2 ± 1.0 |
| | w/ CO | w/o NS | 82.9 ± 0.8 | 66.6 ± 0.7 |
| | | w/ NS | 84.4 ± 0.8 | 68.1 ± 0.9 |
| w/ EM | w/o CO | w/o NS | 84.4 ± 0.0 | 68.8 ± 0.4 |
| | | w/ NS | 85.6 ± 0.3 | 68.7 ± 0.5 |
| | w/ CO | w/o NS | 83.8 ± 0.2 | 67.4 ± 0.4 |
| | | w/ NS | **85.8 ± 0.6** | **69.4 ± 0.3** |

Table 2.1 – Performance of different configurations of our algorithm compared to the results of [Cho, 2015] on Object Discovery and VOC_6x2 datasets in the separate setting. Best results are in bold. We observe that both the normalized score (NS) and the ensemble method (EM) improve the performance. EM also improves the stability of our solution (lower variance). The combination of ensemble method (EM), continuous optimization (CO) and normalized scores (NS) produces the best results for OSD.

The results are shown in Table 2.1. We have found a small bug in the publicly available code of [Cho, 2015], and report both the results from [Cho, 2015] and those we obtained after correction. We observe that the normalized standout score always gives comparable or better results than its unnormalized counterpart, while the ensemble method also improves both the score and the stability (lower variance) of our solution. Combining the normalized standout score, the ensemble method, and the continuous optimization initialization to the greedy ascent yields the best performance. Our best results outperform [Cho, 2015] by small but statistically significant margins: 1.6% for OD and 1.8% for VOC_6x2. Finally, to assess the merit of the continuous optimization, we have measured its duality gap on OD and VOC_6x2: it ranges from 1.5% to 8.7% of the energy, with an average of 5.2% and 3.9% on the two datasets respectively.

We now evaluate our algorithm on VOC_all. As the complexity of solving the max flow problem grows very fast with the number of images, for configurations with continuous optimization, we reduce the number of non-zero entries in each standout matrix such that the total number of nodes in the graph is around $2 \times 10^7$. These standout matrices are then used in rounding the continuous solution, but in the greedy ascent procedure we switch to standout matrices with 1000 non-zero entries. For configurations without the continuous optimization, we always use the standout matrices with 1000 non-zero entries. Also, to reduce the memory footprint of our method, we prefilter the set of potential neighbors of each image for large classes. Pre-filtering is done by marking 100 nearest neighbors of each image in terms of Euclidean distance between

| Method | | VOC_all |
|---|---|---|
| [Cho, 2015] | | 36.6 |
| [Cho, 2015] (our execution) | | 37.6 |
| Ours, w/o CO | w/o EM | 36.4 ± 0.3 |
| | w/ EM | 39.0 ± 0.2 |
| Ours, w/ CO | w/o EM | 37.8 ± 0.3 |
| | w/ EM | 39.2 ± 0.2 |
| [Li, 2016]† | | 40.0 |
| [Wei, 2017a]† | | **46.9** |

Table 2.2 – Performance on VOC_all in separate setting with different configurations of our method compared to baselines. The combination of continuous optimization (CO) and ensemble method (EM) yeilds the best results for our method. Note that [Li, 2016] and [Wei, 2017a] use pre-trained CNN features [Simonyan, 2015a] while [Cho, 2015] and our method use the hand-crafted WHO [Hariharan, 2012] features.

GIST [Torralba, 2008] descriptors as potential neighbors. In the separate setting, we only apply the pre-filtering on the class *person* which has 1023 images. The other classes are sufficiently small for not resorting to the prefiltering procedure.

Table 2.2 shows the CorLoc values obtained by our method with different configurations compared to [Cho, 2015]. We use the normalized score in all of these experiments. It can be seen that the ensemble postprocessing and the continuous optimization are still helpful on this dataset. We obtain the best result with the configuration that includes both of them, which is 1.6% better than [Cho, 2015]. However, our performance is still inferior to state of the art in image colocalization [Li, 2016; Wei, 2017a] which employs deep features from convolutional neural networks trained for image classification and explicitly exploits the single-class assumption.

**Mixed setting.** We now compare in Table 2.3 the performance of our algorithm to [Cho, 2015] in the mixed setting (none of the other methods is applicable to this case). It can be seen that our algorithm without the continuous optimization has the best performance among those in consideration. Compared to [Cho, 2015], it gives a CorLoc 0.8% better on OD dataset, 4.3% better on VOC_6x2 and 2.3% better on VOC_all. The decrease in performance of our method when using the continuous optimization is likely due to the fact that we use standout matrices with only 200 non-zero entries on OD, 100 non-zero entries on VOC_6x2 and VOC_all (due to the limit on the number of nodes of the bipartite graphs) in the configuration with the continuous optimization while we use denser standout matrices (1000 non-zero entries) in the configuration without the continuous optimization.

**Sensitivity to $\nu$.** We compare the performance of our method when using different values of $\nu$ on the VOC_6x2 dataset. Table 2.4 shows the CorLoc obtained by different configurations of our algorithm, all with normalized standout. The performance consistently increases with the value of $\nu$ on this dataset. In all other experiments, however, we set $\nu = 5$ to ease comparisons

| Method | OD | VOC_6x2 | VOC_all |
|---|---|---|---|
| [Cho, 2015] | - | - | 37.6 |
| [Cho, 2015] (our execution) | 82.2 | 55.9 | 37.5 |
| Ours, w/o CO | 83.0 ± 0.4 | 60.2 ± 0.4 | 39.8 ± 0.2 |
| Ours, w/ CO | 80.8 ± 0.5 | 59.3 ± 0.4 | 38.5 ± 0.2 |

Table 2.3 – Performance of our method compared to [Cho, 2015] in the mixed setting.

to [Cho, 2015].

| Method | | | VOC_6x2 |
|---|---|---|---|
| $\nu = 1$ | w/o CO | w/o EM | 63.5 ± 1.2 |
| | | w/ EM | 67.7 ± 0.8 |
| | w/ CO | w/o EM | 65.8 ± 0.8 |
| | | w/ EM | **68.1 ± 0.7** |
| $\nu = 5$ | w/o CO | w/o EM | 67.2 ± 1.0 |
| | | w/ EM | 68.7 ± 0.5 |
| | w/ CO | w/o EM | 68.1 ± 0.9 |
| | | w/ EM | **69.4 ± 0.3** |
| $\nu = 10$ | w/o CO | w/o EM | 68.6 ± 1.0 |
| | | w/ EM | 69.1 ± 0.3 |
| | w/ CO | w/o EM | 68.9 ± 0.7 |
| | | w/ EM | **70.0 ± 0.3** |

Table 2.4 – Performance of different configurations of our algorithm with $\nu = 1$, $\nu = 5$ and $\nu = 10$. Larger values of $\nu$ yield better performance but we use $\nu = 5$ in our experiments to facilitate comparisons to [Cho, 2015].

**Using deep features.** Since activations from deep neural networks trained for image classification (deep features) are known to be better image representations than handcrafted features in various tasks, we have also experimented with such descriptors. We have replaced WHO [Hariharan, 2012] by activations from different layers in VGG16 [Simonyan, 2015a], when computing the appearance similarity $a_{ij}^{kl}$ between regions. In this case, the appearance similarity between two regions is simply the scalar product of the corresponding deep features (normalized or not). As a preliminary experiment to evaluate the effectiveness of deep features, we have run our algorithm without the continuous optimization with the standout score computed using layers *conv4_3*, *conv5_3* and *fc6* in VGG16. Table 2.5 shows the results of these experiments. Surprisingly, most of the tested deep features give worse results than WHO. This may be due to the fact that our matching task is more akin to image retrieval than classification, for which deep features are typically trained. Among those tested, only a variant of the features extracted from the layer *conv5_3* of VGG16 gives an improvement (about 2%) compared to the result obtained by using WHO.

| Features | | | Average |
|---|---|---|---|
| WHO [Hariharan, 2012] | | | 68.8 ± 0.5 |
| *conv4_3* | warping + center cropping | unnormalized | 64.2 ± 0.2 |
| | | normalized | 57.1 ± 0.6 |
| | ROI pooling [Girshick, 2015] | unnormalized | 63.1 ± 0.2 |
| | | normalized | 63.4 ± 0.4 |
| *conv5_3* | warping + center cropping | unnormalized | 64.9 ± 0.2 |
| | | normalized | 64.1 ± 0.4 |
| | ROI pooling [Girshick, 2015] | unnormalized | **70.7 ± 0.2** |
| | | normalized | 68.2 ± 0.3 |
| *fc6* | warping + center cropping | unnormalized | 61.3 ± 0.2 |
| | | normalized | 61.0 ± 0.4 |

Table 2.5 – Performance of our algorithm with deep features on VOC_6x2 in the separate setting.

**Unsupervised initial proposals.** It should be noted that, although our algorithm like that of [Cho, 2015] is totally unsupervised once *given the region proposals*, the randomized Prim's algorithm itself is supervised [Manen, 2013]. To study the effect of this built-in supervision, we have also used the unsupervised *selective search* algorithm [Uijlings, 2013] for generating region proposals. We have conducted experiments on VOC_6x2 dataset with the three different settings of selective search (*fast*, *medium* and *quality*). As one might expect, the *fast* mode gives the smallest number of proposals and of *positive* ones (proposals whose *IoU* with one ground-truth box is at least 0.5). The *quality* mode outputs the largest set of proposals and of positive ones, and the *medium* mode lies in-between. To compare with [Cho, 2015], we also run their public software with each mode of selective search.

| Proposal algorithm | | [Cho, 2015] | Ours |
|---|---|---|---|
| selective search | *fast* | 23.3 | 41.4 ± 0.5 |
| | *medium* | 20.6 | 48.4 ± 0.5 |
| | *quality* | 32.6 | 62.8 ± 0.6 |
| randomized Prim | | 67.6 | 69.4 ± 0.4 |

Table 2.6 – Object discovery on VOC_6x2 with region proposals generated by selective search [Uijlings, 2013] and randomized Prim [Manen, 2013].

The results are shown in Table 2.6. It can be seen that the performance of both [Cho, 2015] and our method drop significantly when using selective search. This may be due to the fact that the percentage of positive proposals found by selective search is much smaller than that of randomized Prim. However, we see that with the *quality* mode of selective search, our method gives results quite close to those of RP, whereas the method in [Cho, 2015] fails badly. This suggests that our method is more robust.

**Visualization.** In order to gain insight into the structures discovered by our approach, we derive from its output a graph of image regions and visualize its main connected components.



Figure 2.2 – Visualization of VOC_6x2 in the mixed setting. The figure shows the third component in the graph of regions, corresponding roughly to class *motorbike*. The two first components are shown in Figure 2.1.

The nodes of this graph are the image regions that have been finally retained. Two regions $(i, k)$ and $(j, l)$ are connected if the images containing them are neighbors in the discovered undirected image graph ($e_{ij}$ or $e_{ji} = 1$) and the standout score between them, $S_{ij}^{kl}$, is greater than a certain threshold.

Choosing the threshold to get a sufficient number of large enough components for visualization purpose has proven difficult. We used instead an iterative procedure: the graph is first constructed with a high threshold to produce a small number of connected components of reasonable size, which are removed from the graph. On the remaining graph, a new, suitable threshold is found to get new components of sufficient size. This is repeated until a target number of components is reached.

When applied to our results in the mixed setting on VOC_6x2 dataset, this visualization procedure yields clusters that roughly match object categories. In Figure 2.1, we show sub-sampled graphs (for visualization purpose) of the two first components, which roughly correspond to classes *bicycle* and *aeroplane*. The third component is shown in Figure 2.2. Although containing also images of other classes, it is by far dominated by *motorbike* images. The visualization suggests that our model does extract meaningful semantic structures from the image collections and regions they contain.

## 2.5 Conclusion and Limitations

We have presented an optimization-based approach to fully unsupervised image matching and object discovery, and demonstrated its promise on several standard benchmarks. In its current form, the algorithm has some limitations. First, due to high computational and memory cost, it is limited to relatively small datasets. Second, although OSD works with any type of regions proposals, it obtains the best results with the supervised randomized Prim proposals while producing unsatisfactory performance with *fast* selective search, a type of unsupervised regions proposals which yields the same computational complexity in OSD. Third, CNN features, which often boost the performance of other tasks significantly when replacing hand-crafted features, do not yield similar improvements in our experiments for object discovery. Finally, similar to [Cho, 2015], OSD discovers only one object per image, limiting its applications on complex, natural images. In the next chapter, we will address all of these issues.

# Chapter 3

# Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections

Objectives

In this chapter, we build on the optimization approach of OSD in Chapter 2 with several key novelties : (1) We propose a novel saliency-based region proposal algorithm that achieves significantly higher overlap with ground-truth objects than other competitive methods. This procedure leverages off-the-shelf CNN features trained on classification tasks without any bounding box information, but is otherwise unsupervised. (2) We exploit the inherent hierarchical structure of our region proposals as an effective regularizer for OSD, boosting its performance to significantly improve over the state of the art on several standard benchmarks, and enabling for the first time (to the best of our knowledge) the discovery of multiple objects per image. (3) We adopt a two-stage strategy to first select promising proposals using small random sets of images before using the whole image collection to discover the objects it depicts, allowing us to tackle datasets with up to 20,000 images, an over five-fold increase compared to OSD, and a first step toward true large-scale unsupervised image interpretation.
This work, done in collaboration with Patrick Pérez and Jean Ponce, has appeared in Proceedings of the European Conference on Computer Vision (ECCV) 2020.

**Contents**


3.1 Introduction . . . . . **38**
3.2 Related Work . . . . . **40**
3.3 Proposed Approach . . . . . **41**
3.3.1 Region Proposals from CNN Features . . . . . 41
3.3.2 Regularized OSD . . . . . 42
3.3.3 Large-Scale Object Discovery . . . . . 43
3.4 Experiments . . . . . **46**
3.4.1 Experimental Setup . . . . . 46
3.4.2 Implementation Details . . . . . 46
3.4.3 Region Proposal Evaluation . . . . . 48
3.4.4 Object Discovery Performance . . . . . 49
3.5 Conclusion and Limitations . . . . . **54**

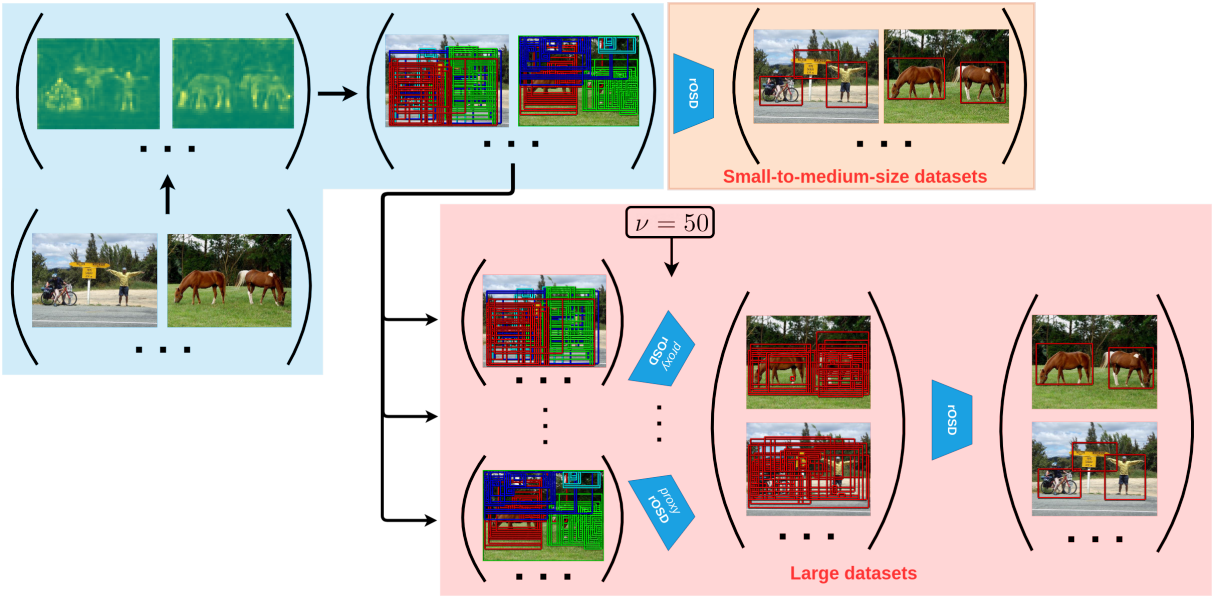



Figure 3.1 – An overview of the different modules of the proposed rOSD method. Given an image collection, we first extract CNN features and generate region proposals for each image (blue zone). These proposals can be divided into disjoint groups, each corresponding to at most one object and shown with a different color. We then run rOSD directly on the collection if it is not too large (orange zone) or run the two-stage large-scale algorithm (red zone) otherwise.

## 3.1 Introduction

In this chapter, we introduce regularized OSD, or rOSD, a method for unsupervised object discovery built on the OSD framework introduced in the previous chapter. rOSD aims to alleviate the limitations of OSD and improves it to effectively discover multiple objects in large image collections. Let us first provide a short recap of OSD and discuss its limitations.

Given a collection of $n$ images, possibly containing objects from different categories, each equipped with $p$ region proposals (which can be obtained using selective search [Uijlings, 2013], edgeboxes [Zitnick, 2014], randomized Prim [Manen, 2013], etc.) and a set of potential neighbors for each image, the unsupervised object and structure discovery problem is formalized in OSD as a discrete optimization problem over a set of variables that describe the structure of the implicit graph of images. Let us define the variable $e$ as an element of $\{0,1\}^{n\times n}$ with a zero diagonal, such that $e_{ij} = 1$ when images $i$ and $j$ are linked by a (directional) edge in the implicit graph of images, and $e_{ij} = 0$ otherwise, and the variable $x$ as an element of $\{0,1\}^{n\times p}$, with $x_i^k = 1$ when region proposal number $k$ of image $i$ corresponds to visual content shared with its neighbors in the graph. As noted in Chapter 2, this leads to the following optimization problem:

$$\max_{x,e} S(x,e) = \sum_{i=1}^{n} \sum_{j\in N(i)} e_{ij} x_i^T S_{ij} x_j, \text{ s.t. } \sum_{k=1}^{p} x_i^k \leq \nu \text{ and } \sum_{j\neq i} e_{ij} \leq \tau \ \forall i, \tag{3.1}$$

where $N(i)$ is the set of potential neighbors of image $i$, $S_{ij}$ is a $p \times p$ matrix whose entry $S_{ij}^{kl}$

measures the similarity between regions $k$ and $l$ of images $i$ and $j$, and $\nu$ and $\tau$ are predefined constants corresponding respectively to the maximum number of objects present in an image and to the maximum number of neighbors an image may have. This is a hard combinatorial optimization problem. As shown in Chapter 2, an approximate solution can be found by (a) a dual gradient ascent algorithm for a continuous relaxation of Equation (3.1) with exact updates obtained by maximizing a supermodular cubic pseudo-Boolean function [Bach, 2013; Nedić, 2009], (b) a simple greedy scheme, or (c) a combination thereof. Since solving the continuous relaxation of Equation (3.1) is computationally expensive and may be less effective for large datasets (see Chapter 2), we only consider the version (b) of OSD in our analysis.

OSD has some limitations: (1) Although the algorithm itself is fully unsupervised, it gives by far its best results with region proposals from randomized Prim [Manen, 2013], a region proposal algorithm trained with bounding box supervision. (2) Whitened HOG (WHO) [Hariharan, 2012] is used to represent region proposals in OSD although CNN features work better on the similar image colocalization problem [Li, 2016; Wei, 2019]. Naively switching to CNN features does not give consistent improvement on common benchmarks. OSD with CNN features gives a CorLoc of 82.9, 71.5 and 42.8 compared to 87.1, 71.2 and 39.5 given by OSD with WHO [1], respectively on OD, VOC_6x2 and VOC_all data sets. (3) Finally, due to its high memory cost, the algorithm cannot be applied to large datasets without compromising its final performance. In the next sections, we describe our approach to addressing these limitations, as well as extending OSD to solve multi-object discovery.

Our contributions in this chapter can be summarized as follows:

- We propose a simple but effective method for generating region proposals directly from CNN features (themselves trained beforehand on some auxiliary task [Simonyan, 2015a] *without* bounding boxes) in an unsupervised way (Section 3.3.1). Our algorithm gives on average half the number of region proposals per image compared to selective search [Uijlings, 2013], edgeboxes [Zitnick, 2014] or randomized Prim [Manen, 2013], yet significantly outperforms these off-the-shelf region proposals in object discovery (Table 3.2).
- Leveraging the intrinsic structure of region proposals generated by our method allows us to add an additional constraint into the OSD formulation that acts as a regularizer on its behavior (Section 3.3.2). This new formulation, rOSD, significantly outperforms the original algorithm and allows us to effectively perform multi-object discovery, a setting never studied before (to the best of our knowledge) in the literature.
- We propose a two-stage algorithm to make rOSD applicable to large image collections (Section 3.3.3). In the first stage, rOSD is used to choose a small set of good region proposals for each image. In the second stage, these proposals and the full image collection are fed to rOSD to find the objects and the image graph structure.
- We demonstrate that our approach yields significant improvements over the state of the art in object discovery at the time of its writing (Tables 3.3 and 3.4). We also run our two-stage algorithm on a new dataset with about 20,000 images, which is much larger than

1. We use here a symmetrized version of the similarity score matrices $S_{ij}$ which yields slightly better results for OSD than the original results in Chapter 2.

the VOC_all dataset considered in OSD, and show that it significantly outperforms the plain OSD in this setting (Table 3.6).

An overview of rOSD is given in Figure 3.1. The only supervisory signal used in our setting are the image labels used to train CNN features in an auxiliary classification task (see [Li, 2016; Wei, 2019] for similar approaches in the related colocalization domain). We use CNN features trained on ImageNet classification [Simonyan, 2015a], *without* any bounding box information. Our region proposal and object discovery algorithms are otherwise fully unsupervised.

## 3.2 Related Work

**Region proposals** have been used in object detection/discovery to serve as object priors and reduce the search space. In most cases, they are found either by a bottom-up approach in which low-level cues are aggregated to rank a large set of boxes obtained with sliding window approaches [Alexe, 2012; Uijlings, 2013; Zitnick, 2014] and return the top windows as proposals, or by training a model to classify them (as in randomized Prim [Manen, 2013], see also [Ren, 2015a]), with *bounding box supervision*. Edgeboxes [Zitnick, 2014] and selective search [Uijlings, 2013] are popular off-the-shelf algorithms that are used to generate region proposals in object detection [Girshick, 2014; Girshick, 2015], weakly-supervised object detection [Cinbis, 2017; Tang, 2018a] or image colocalization [Li, 2016]. Note, however, that the features used to generate proposals in these algorithms and those representing them in the downstream tasks are generally different in nature: Typically, region proposals are generated from low-level features such as color and texture [Uijlings, 2013] or edge density [Zitnick, 2014], but CNN features are used to represent them in downstream tasks. However, the region proposal network in Faster-RCNN [Ren, 2015a] shows that proposals generated directly from the features used in the object detection task itself give a significant boost in performance. In the object discovery setting, we therefore propose a novel approach for generating region proposals in an unsupervised way from CNN features trained on an auxiliary classification task without bounding box information.

Features from CNNs trained on large-scale image classification have also been used to localize object in the weakly-supervised setting. [Zhou, 2016] and [Selvaraju, 2017] fine-tune a pre-trained CNN to classify images and construct class activation maps, as weighted sums of convolutional feature maps or their gradient with respect to the classification loss, for localizing objects in these images. [Tang, 2018b] generates region proposals to perform weakly-supervised object detection on a set of labelled images by training a proposal network using the image labels as supervision. Contrary to these works, we generate region proposals using only pre-trained CNN features *without* fine-tuning the feature extractor. Moreover, our region proposals come with a nice intrinsic structure which can be exploited to boost object discovery performance.

## 3.3 Proposed Approach

### 3.3.1 Region Proposals from CNN Features

We address the limitation of using off-the-shelf region proposals in OSD with insights gained from the remarkably effective method for image colocalization proposed by [Wei, 2019]: CNN features pre-trained for an auxiliary task, such as ImageNet classification, give a strong, *category-independent* signal for unsupervised tasks. In retrospect, this insight is not particularly surprising, and it is implicit in several successful approaches to image retrieval [Zhang, 2015a] or co-saliency detection [Babenko, 2014; Babenko, 2015; Wei, 2017b; Hsu, 2018b]. [Wei, 2019] uses it to great effect in the image colocalization task. Feeding an image to a pre-trained convolutional neural network yields a set of feature maps represented as a 3D tensor (e.g., a convolutional layer of VGG16 [Simonyan, 2015a] or ResNet [He, 2016]). [Wei, 2019] observes that the "image" obtained by simply adding the feature maps gives hints to the locations of the objects it contains, and identifies objects by clustering pixels with high activation. Similar but different from them, we observe that local maxima in the above "images" correspond to salient parts of objects in the original image and propose to exploit this observation for generating region proposals directly from CNN features. As we do not make use of any annotated bounding boxes, our region proposal itself is indeed unsupervised.

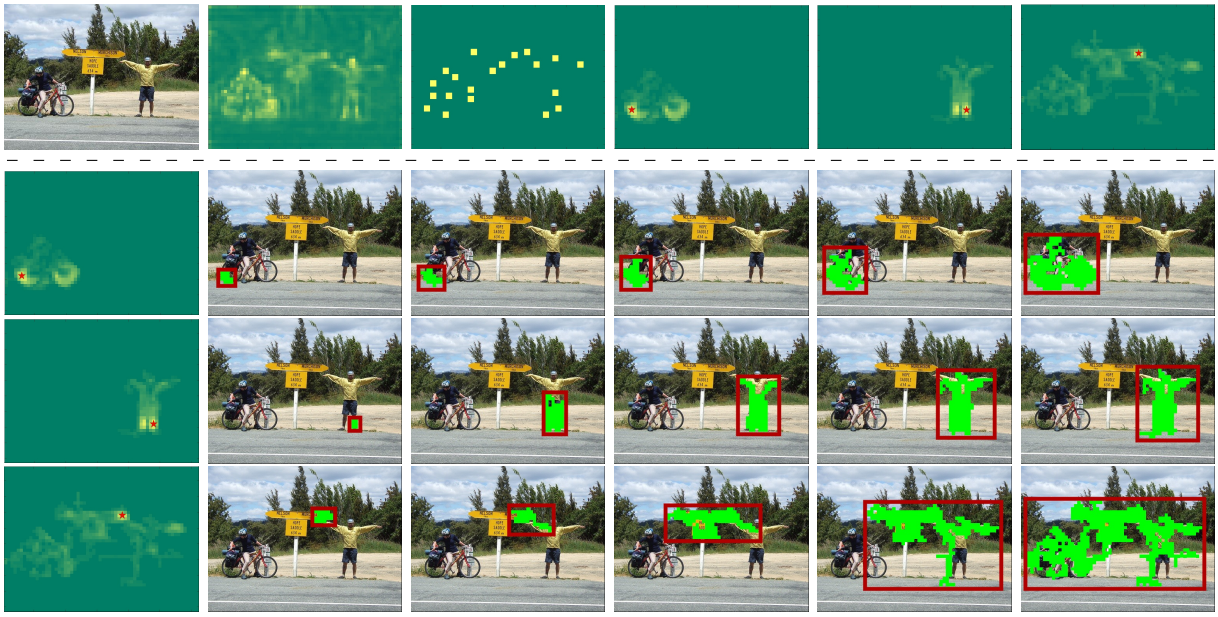


Figure 3.2 – Illustration of the unsupervised region proposal generation process. The top row shows the original image, the global saliency map $s_g$, local maxima of $s_g$ and three local saliency maps $s_y$ from three local maxima (marked by red stars). The next three rows illustrate the proposal generation process on the local saliency maps: From left to right, we show in green the connected component formed by pixels with saliency above decreasing thresholds and, in red, the corresponding region proposals.

Our method consists of the following steps. First, we feed the image to a pre-trained convolu-

tional neural network to obtain a 3D tensor of size $(H \times W \times D)$, noted $F$. Adding elements of the tensor along its depth dimension yields a $(H \times W)$ 2D saliency map, noted as $s_g$ (*global* saliency map), showing salient locations in the image with each location in $s_g$ being represented by the corresponding $D$-dimensional feature vector from $F$. Next, we find robust local maxima in the previous saliency map using *persistence*, a measure used in topological data analysis [Edelsbrunner, 2002; Zomorodian, 2005; Edelsbrunner, 2009; Chazal, 2013; Oudot, 2015] to find significant critical points of a function (see Section 3.4.2 for details). We find regions around each local maximum $y$ using a *local* saliency map $s_y$ of the same size as the global one. The value at any location in $s_y$ is the dot product between normalized feature vectors at that location and at the local maximum. By construction, the local saliency map highlights locations that are likely to belong to the same object as the corresponding local maximum. Finally, for each local saliency map, we discard all locations with scores below some threshold and the bounding box around the connected component containing the corresponding local maximum is returned as a region proposal. By varying the threshold, we can obtain tens of region proposals per local saliency map. An example illustrating the whole process is shown in Figure 3.2.

### 3.3.2 Regularized OSD

Due to its greedy nature, the block-coordinate ascent algorithm we use to solve OSD (Algorithm 2.1) is prone to bad local maxima. This problem can be partially resolved by using a larger value of $\nu$ in the optimization than the actual number of objects we intend to retrieve (which is one in OSD) to diversify the set of retained regions in each iteration. Amongst regions retained in each image, a single one is then selected in a post processing step by ranking these using a new score solely based on their similarity to the retained regions in the image's neighbors (see Section 2.2.6 in Chapter 2). Increasing $\nu$ in fact gives limited help in diversifying the set of retained regions. Since there is redundancy in object proposals with many highly overlapping regions, the $\nu$ retained regions are often nearly identical (Figure 3.3, second row). This phenomenon also prevents OSD from retrieving multiple objects in images. One can use the ranking in OSD's post processing step with non-maximum suppression to return more than one region from $\nu$ retained regions but since $\nu$ regions are often highly overlapping, this fails to localize multiple objects.

By construction, proposals produced by our approach also contain many highly overlapping regions, especially those generated from the same local maximum in the saliency map. However, they come with a nice intrinsic structure: Proposals in an image can be partitioned into groups labelled by the local maximum from which they are generated. Naturally, it makes sense to impose that at most one region in a group is retained in OSD since they are supposed to correspond to the same object. This additional constraint also conveniently helps to diversify the set of proposals returned by the block-coordinate ascent procedure by avoiding to retain highly overlapping regions (Figure 3.3, third row). Concretely, let $G_{ih}$ be the set of region proposals in image $i$ generated from the $h$-th local maximum of its global saliency map $s_g$, with $1 \leq g \leq L_i$ where $L_i$ is the number of local maxima in $s_g$; we propose to add the corresponding

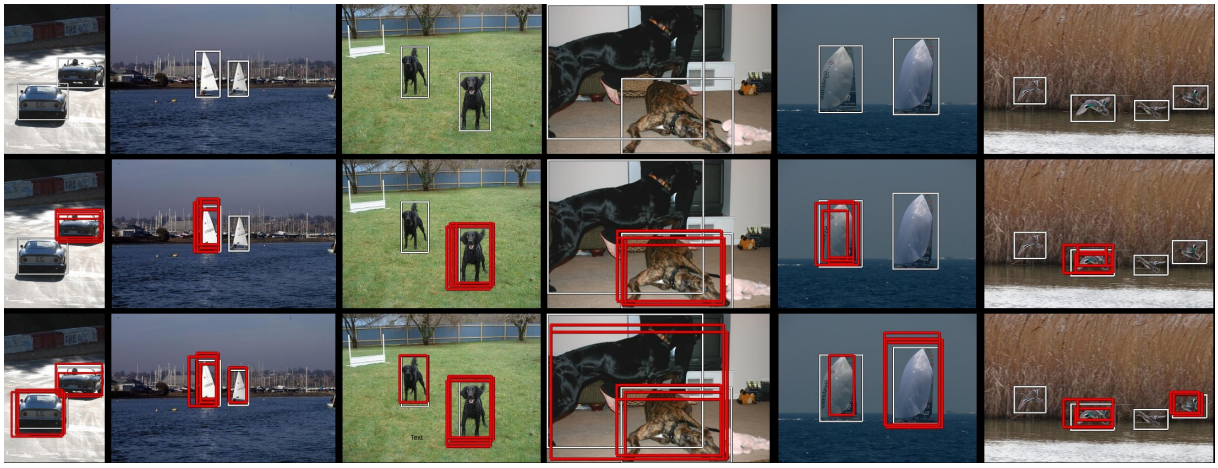


Figure 3.3 – Regions returned by OSD and rOSD. In each column from top to bottom: original image, image with regions returned by OSD, image with regions returned by rOSD. OSD tends to returns nearly identical regions around a single object while rOSD selects a more diverse set of regions.

constraints to Equation (3.1):

$$\max_{x,e} S(x,e) = \sum_{i=1}^{n} \sum_{j \in N(i)} e_{ij} x_i^T S_{ij} x_j, \text{ s.t.} \forall i \begin{cases} \sum_{k=1}^{p} x_i^k \leq \nu, \\ \sum_{k \in G_{ih}} x_i^k \leq 1, \text{ for all groups } h \\ \sum_{j \neq i} e_{ij} \leq \tau. \end{cases} \tag{3.2}$$

We coin the new formulation *regularized* OSD, or rOSD. A solution to rOSD can be obtained by a greedy block-coordinate ascent algorithm, similar to Algorithm 2.1 in Chapter 2 but slightly modified to account for the new constraints. Its iterations are illustrated in Algorithm 3.1.

Note that we do not use the convex optimization technique in Chapter 2 to solve rOSD due to its high computational cost. We will demonstrate the effectiveness of rOSD compared to OSD and the state of the art in Section 3.4.4.

### 3.3.3 Large-Scale Object Discovery

The optimization algorithm for OSD [2] requires loading all score matrices $S_{ij}$ into memory (they can also be computed on-the-fly but at an unacceptable computational cost). The corresponding memory cost is $M = (\sum_{i=1}^{n} |N(i)|) \times K$, decided by two main factors: The number of image pairs considered $\sum_{i=1}^{n} |N(i)|$ and the number of positive entries $K$ in matrices $S_{ij}$. To reduce the cost on larger datasets, we pre-filter the neighborhood of each image ($|N(i)| \leq 100$ for classes with more than 1000 images) and limit $K$ to 1000. This value of $K$ is approximately the average number of proposals in each image, and it is intentionally chosen to make sure that $S_{ij}$ is not too sparse in the sense that approximately every proposal in image $i$ should have a

2. Since the analysis in this section applies to both OSD and rOSD, we refer to both as OSD for ease of notation.

**Algorithm 3.1:** Block coordinate ascent algorithm for rOSD.

**Input:** Groups $G_i$, parameters $\nu$ and $\tau$, score matrices $S_{ij}$, number $n$ of images.
**Result:** A solution to rOSD.

$x_i \leftarrow \mathbb{1}_p \; \forall i$, $e_{ij} \leftarrow 1 \; \forall i \neq j$ ▷ Initialize $x$ and $e$
$\sigma \leftarrow$ `rand-perm`$([1..n])$ ▷ Generate a random permutation of $[1..n]$
**for** $i = 1$ *to* $n$ **do**
  $C_x \leftarrow \sum_{j \neq \sigma(i)}^{n} (e_{\sigma(i)j} S_{\sigma(i)j} + e_{j\sigma(i)} S_{j\sigma(i)}^T) x_j$ ▷ Comp. coef. in $S(x, e)$ of elem. of $x_{\sigma(i)}$
  $I \leftarrow \emptyset$
  **for** $h = 1;\ h \leq L_i$ **do**
    $r^* \leftarrow \arg\max_{r \in G_{ih}} C_x(r)$ ▷ Find reg. w. greatest score in $h$-th group
    $I \longleftarrow I \cup \{r^*\}$.
  **end**
  $k_1, \ldots, k_\nu \leftarrow$ `find-max-indices`$(C_x, I)$ ▷ Find within $I$ ids of $\nu$ largest elem. in $C_x$
  $x_{\sigma(i)} \leftarrow 0$ ▷ Re-initialize $x_{\sigma(i)}$
  **for** $t = 1$ to $\nu$ **do**
    $x_{\sigma(i)}^{k_t} \leftarrow 1$ ▷ Assign elements with indices $k_1, \ldots, k_\nu$ to 1
  **end**
**end**
**for** $i = 1$ to $n$ **do**
  $C_e \leftarrow [x_i^T S_{i1} x_1; x_i^T S_{i2} x_2; \ldots; x_i^T S_{in} x_n]$ ▷ Comp. coef. in $S(x, e)$ of elem. in $e_i$
  $j_1, \ldots, j_\tau \leftarrow$ `find-max-indices`$(C_e, \tau)$ ▷ Find ids of $\tau$ largest elem. in $C_e$
  $e_i \leftarrow 0$ ▷ Re-initialize $e_i$
  **for** $t = 1$ to $\tau$ **do**
    $e_{ij_t} \leftarrow 1$ ▷ Assign elements with indices $j_1, \ldots, j_\tau$ to 1
  **end**
**end**

positive match with some proposal in image $j$. Further reducing the number of positive entries in score matrices is likely to hurt the performance (see Table 3.6) while a number of 100 potential neighbors is already small and can not be significantly lowered. Effectively scaling up OSD therefore requires lowering considerably the number of proposals it uses. To this end, we propose two different interpretations of the image graph $G = (x, e)$ obtained by solving Equation (3.1) and exploit both to scale up OSD.

**Two different interpretations of the image graph.** $G$ can be interpreted as capturing the "true" structure of the input image collection. In this case, $\nu$ is typically small (say, 1 to 5) and the discovered "objects" correspond to maximal cliques of $G$, with instances given by active regions ($x_i^k = 1$) associated with nodes in the clique. But it can also be interpreted as a *proxy* for that structure. In this case, we typically take $\nu$ larger (say, 50). The active regions found for each node $x_i$ of $G$ are interpreted as the most promising regions in the corresponding image and the active edges $e_{ij}$ link it to other images supporting that choice. We dub this variant *proxy* OSD.

For small image collections, it makes sense to run OSD only. For large ones, we propose instead to split the data into random groups with roughly equal size, run proxy OSD on each

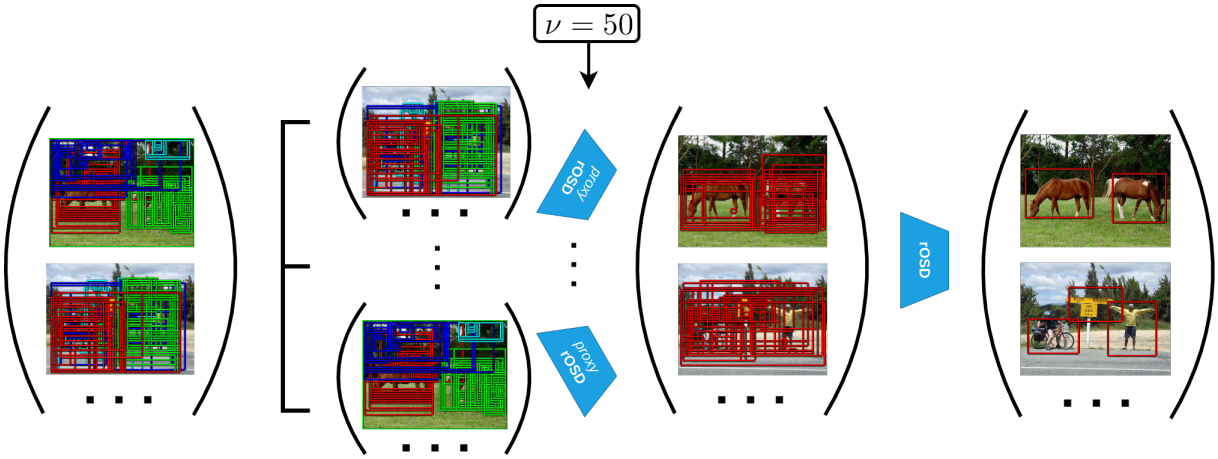



Figure 3.4 – An illustration of the proposed two-stage algorithm for large-scale unsupervised object discovery. We first run proxy rOSD in parallel on different parts of an partition of the image collection to select approximately $\nu$ good regions for each image. In the second stage, we then run rOSD on the entire image set using only the selected region proposals.

group to select the most promising region proposals in the corresponding images, then run OSD using these proposals on the entire image collection. Using this two-stage algorithm, we reduce significantly the number of image pairs in each run of the first stage, thus allowing the use of denser score matrices in these runs. In the second stage, since only a very small number of region proposals is considered in each image, we need to keep only a few positive entries in each score matrix and are able to run OSD on the entire image collection. Our approach for large-scale object discovery is summarized in Figure 3.4 and Algorithm 3.2.

**Algorithm 3.2:** Large-scale object discovery algorithm.

**Input:** Dataset $D$ of $n$ images, memory limit $M$, number of partition $k$, image neighborhood size $N_b$, parameters $\nu^*$ and $\tau$.

Partition $D$ into random $k$ parts $D_1$, ..., $D_k$, each has roughly $\lfloor n/k \rfloor$ images.
Compute the maximum number of positive entries in the score matrices in each part: $K_1 \longleftarrow M/(N_b \times \lfloor n/k \rfloor)$.
Compute the maximum number of positive entries in the score matrices in the whole dataset: $K_2 \longleftarrow M/(N_b \times n)$.
**for** $i = 1$ *to* $k$ **do**
    Compute score matrices for image pairs in $D_i$ with $K_1$ positive entries.
    Run proxy OSD on $D_i$ with $\nu = K_2$.
    Each image in $D_i$ has a new set of region proposals which are those retained by OSD.
**end**
Compute score matrices between pairs of images in $D$ with $K_2$ positive entries.
Run OSD on the whole dataset $D$ with $\nu = \nu^*$.

## 3.4 Experiments

### 3.4.1 Experimental Setup

**Datasets.** Other than the three datasets – OD, VOC_6x2 and VOC_all – considered in OSD, we additionally evaluate rOSD on VOC12 and C20K datasets. VOC12 is a subset of the PASCAL VOC 2012 dataset [Everingham, 2012] and obtained in the same way as VOC_all: First, images containing only *difficult* or *truncated* objects are eliminated, then objects marked as *difficult* or *truncated* in the remaining images are also dropped. The resulting dataset contains 7838 images and figures 13957 objects. For large-scale experiments, we randomly choose 20000 images from the training set of COCO [Lin, 2014] and eliminate those containing only *crowd* bounding boxes as well as bounding boxes marked as *crowd* in retained images, resulting in the C20K dataset, which has 19817 images and 143951 objects.

**Evaluation Protocols and Metrics.** Different from OSD, we consider in this chapter both the single-object discovery setting, in which the model returns a single object for each image, and the multi-object discovery setting where the model returns possibly more than one object per image. In the single-object setting, similar to OSD, [Cho, 2015] and baselines in colocalization [Li, 2016; Wei, 2019], we use CorLoc as the evaluation metric. In the multi-object setting, since CorLoc does not take into account multiple detections per image, we use instead *detection rate* at the IoU threshold of 0.5 as measure of performance. Given some threshold $\zeta$, detection rate at $IoU = \zeta$ is the percentage of ground-truth bounding boxes that have an $IoU$ with one of the predicted objects at least $\zeta$. We evaluate our method in both the *colocalization* task, where the algorithm is run separately on each class of the dataset, and the average CorLoc/detection rate over all classes is computed as the overall performance measure on the dataset, and the true *discovery* task where the whole dataset is considered as a single class.

### 3.4.2 Implementation Details

**Features.** We test our methods with the pre-trained CNN features from VGG16 and VGG19 [Simonyan, 2015a]. For generating region proposals, we apply the algorithm described in Section 3.3.1 separately to the layers right before the last two max pooling layers of the networks (*relu4_3* and *relu5_3* in VGG16, *relu4_4* and *relu5_4* in VGG19), then fuse proposals generated from the two layers as our final set of proposals. Note that using CNN features at multiple layers is important as different layers capture different visual patterns in images [Zeiler, 2014]. One could also use more layers from VGG16 (e.g., layers *relu3_3*, *relu4_2* or *relu5_2*) but we only use two for efficiency. In experiments with OSD and rOSD, we extract features for the region proposals by applying the region of interest pooling (RoI pooling) operator introduced in Fast-RCNN [Girshick, 2015] to layer *relu5_3* of VGG16 or layer *relu5_4* of VGG19.

**Region proposal generation process.** For finding robust local maxima of the global saliency maps $s_g$, we rank its locations using persistence [Chazal, 2013; Edelsbrunner, 2009; Edelsbrunner, 2002; Oudot, 2015; Zomorodian, 2005]. Concretely, we consider $s_g$ as a 2D image and each

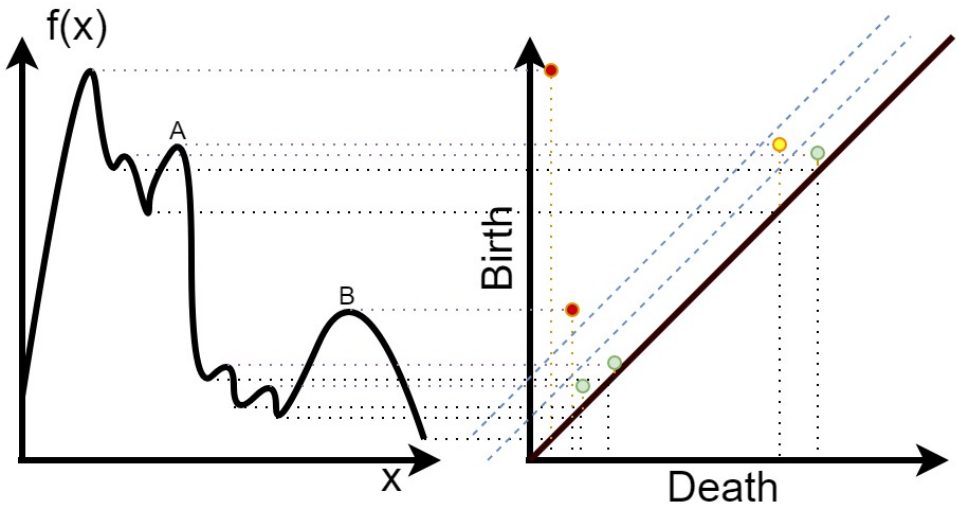
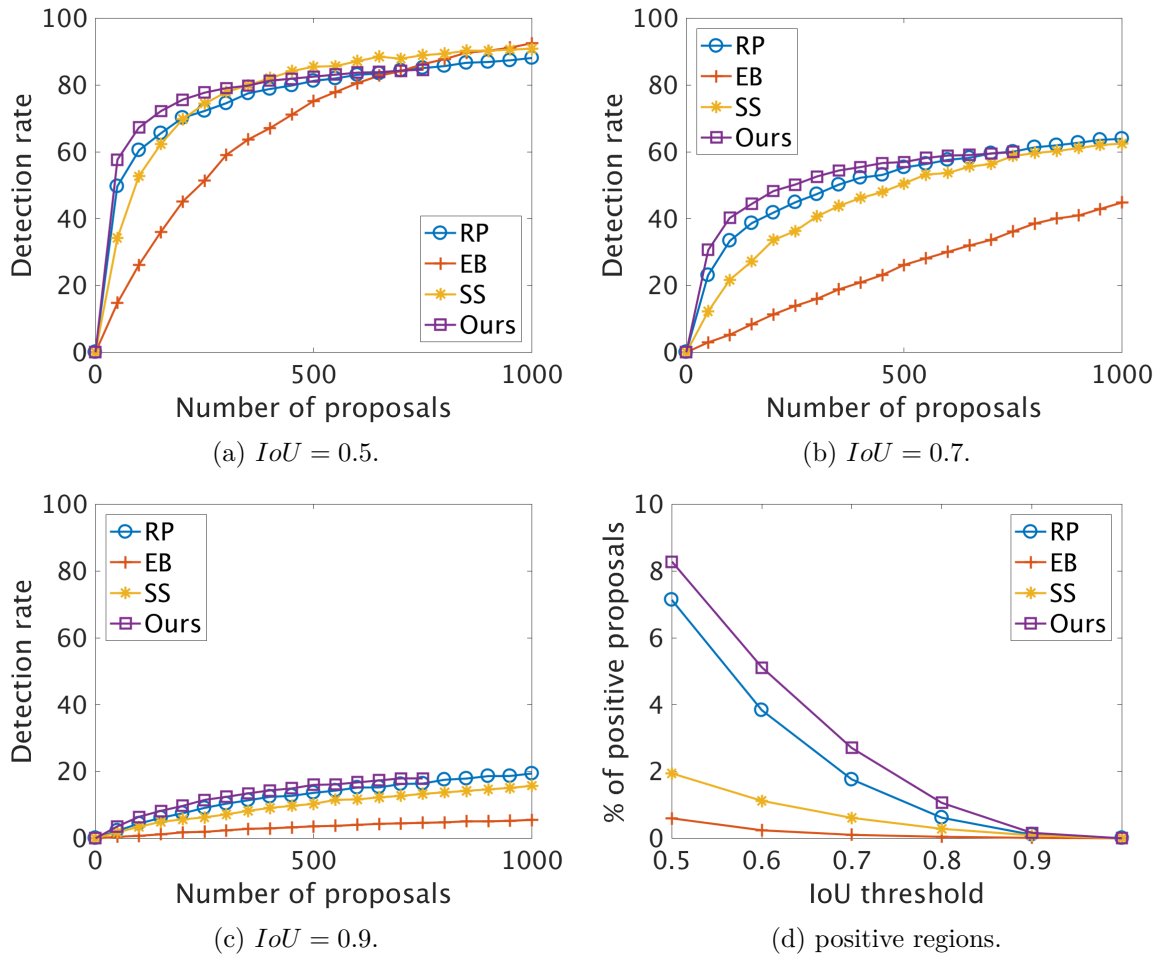



Figure 3.5 – An illustration of persistence in the 1D case. Left: A 1D function. Right: Its persistence diagram. Points above the diagonal correspond to its local maxima and the vertical distance from these points to the diagonal is their persistence. Local maxima with higher persistence are more robust: $B$ is more robust than $A$ although $f(A) > f(B)$. Given a chosen persistence threshold (shown by dash lines in blue), points with persistence higher than some threshold are selected as robust local maxima. The black horizontal dotted lines show birth and death time of the local maxima of $f$.

location in it as a pixel. We associate with each pixel a "birth" time (its own saliency) and a "death" time, defined as the highest value $\eta$ for which there is a path in the 4-neigborhood graph of pixels that connects it with another pixel with higher saliency such that the saliency of all pixels in the path is at least $\eta$, or, if no such path exists, the lowest saliency value in the map. The persistence of a pixel is defined as the difference between its birth and death times. A sorted list of pixels in decreasing persistence order is computed, and the local maxima are chosen as the top pixels in the list. An illustration of this computation for 1D case is illustrated in Figure 3.5.

For additional robustness, we also apply non-maximum suppression on the list over a $3 \times 3$ neighborhood. Since the saliency map created from CNN feature maps can be very noisy, we eliminate locations with score in $s_g$ below $\alpha \max s_g$ before computing the persistence to obtain only good local maxima. When generating proposals from local saliency map $s_y$, we also eliminate locations with score smaller than the average score in $s_y$ and whose score in $s_g$ is smaller than $\beta$ times the average score in $s_g$. We choose the value of the pair $(\alpha, \beta)$ in $\{0.3, 0.5\} \times \{0.5, 1\}$ by conducting small-scale object discovery on VOC_6x2. We find that $(\alpha, \beta) = (0.3, 0.5)$ yields the best performance and gives local saliency maps that are not fragmented while eliminating well irrelevant locations across settings and datasets. We take up to 20 local maxima (after non-maximum suppression) and use 50 linearly spaced thresholds between the lowest and the highest scores in each local saliency map to generate proposals. We study the influence of these parameters in Table 3.1.




(a) $IoU = 0.5$. (b) $IoU = 0.7$.

(c) $IoU = 0.9$. (d) positive regions.

Figure 3.6 – Quality of proposals by different methods. (a-c): Detection rate by number of proposals at different $IoU$ thresholds of randomized Prim (RP) [Manen, 2013], edgeboxes (EB) [Zitnick, 2014], selective search (SS) [Uijlings, 2013] and ours; (d): Percentage of positive proposals for the four methods.

**Object discovery experiments.** For single-object colocalization and discovery, following OSD, we use $\nu = 5, \tau = 10$ and apply the post processing of OSD to obtain the final result. For multi-object setting, we use $\nu = 50$, $\tau = 10$ and apply the post processing with non-maximum suppression at $IoU = 0.7$ to retain at most 5 regions in the final result. Similar to OSD, we report the average performance and its standard deviation from multiple runs for each experiment. However, we do not use the ensemble method (EM) since it does yield clear benefits (see Table 3.7). On large classes/datasets, we pre-filter the set $N(i)$ of neighbors that are considered in the optimization for each image, using the cosine similarity between features from the fully connected layer *fc6* of the pre-trained network, following [Babenko, 2014]. The number of potential neighbors of each image is fixed to 50 in all experiments where the pre-filtering is necessary.

### 3.4.3 Region Proposal Evaluation

Following other works on region proposals [Manen, 2013; Uijlings, 2013; Zitnick, 2014], we evaluate the quality of our proposals on PASCAL VOC 2007 using *detection rate* at various $IoU$ thresholds. But since we intend to later use our proposals for object discovery, unlike other works, we evaluate directly our proposals on VOC_all instead of the test set of VOC 2007 to reveal the link between the quality of proposals and the object discovery performance. Figure 3.6(a-c) shows the performance of different proposals on VOC_all. It can be seen that our method performs better than others at a very high overlap threshold (0.9) regardless of the number of proposals allowed. At medium threshold (0.7), our proposals are on par (or better for fewer than 500 proposals) with those from selective search [Uijlings, 2013] and randomized Prim [Manen, 2013] and much better than those from edgeboxes [Zitnick, 2014]. At a small threshold (0.5), our method is still on par with randomized Prim and edgeboxes, but does not fare as well as selective search. It should be noted that randomized Prim is supervised whereas the others are unsupervised.

In OSD and rOSD, localizing an object in an image means singling out a *positive* proposal, that is, a proposal having an $IoU$ greater than some threshold with object bounding boxes. It is therefore easier to localize the object if the percentage of positive region proposals is larger. As shown by Figure 3.6(d), our method performs very well according to this criterion: Over 8% of our proposals are positive at an $IoU$ threshold of 0.5, and over 3% are still positive for an $IoU$ of 0.7. Also, randomized Prim and our method are by far better than selective search and edgeboxes, which explains the superior object discovery performance of the former over the latter (see Chapter 2 and Table 3.2). Note that region proposals with a high percentage of positive ones could also be used in other tasks, i.e., weakly-supervised object detection, but this is left for future work.

### 3.4.4 Object Discovery Performance

**Single-object colocalization and discovery.** An important component of OSD is the similarity model used to compute score matrices $S_{ij}$. We introduce in Chapter 2 two scores, *confidence* score and *standout* score, but use only the latter for it gives better performance. Since our new proposals come with different statistics, we test both scores with them. Table 3.1 (left) compares colocalization performance on OD, VOC_6x2 and VOC_all of OSD using the confidence and standout scores as well as our proposals. It can be seen that on VOC_6x2 and VOC_all, the confidence score does better than the standout score, while on OD, the latter does better. This is not particularly surprising since images in OD generally contain bigger objects (relative to image size) than those in the other datasets. In fact, although the standout score is used on all datasets in [Cho, 2015] and OSD, the parameter $\gamma$ (see [Cho, 2015]) used in computing the standout score is adjusted to favor larger regions when running on OD. In all of our experiments from now on in this chapter, we use the standout score on OD and the confidence score on other datasets (VOC_6x2, VOC_all, VOC12 and C20K).

Our proposal generation process introduces a few hyper-parameters. Apart from $\alpha$ and

| Dataset | Confidence | Standout |
|---|---|---|
| OD | 83.7 ± 0.4 | **89.0 ± 0.6** |
| VOC_6x2 | **73.6 ± 0.6** | 64.1 ± 0.3 |
| VOC_all | **44.7 ± 0.3** | 41.4 ± 0.1 |

| $(u, v)$ | (20,50) | (20,100) | (50,50) | (50,100) |
|---|---|---|---|---|
| CorLoc | 73.6 ± 0.8 | 73.4 ± 0.7 | 73.3 ± 1.1 | 74.2 ± 0.8 |
| $p$ | 760 | 882 | 1294 | 1507 |

Table 3.1 – Left: Colocalization performance with our proposals in different configurations of OSD. Right: Colocalization performance for different values of hyper-parameters.

$\beta$, two other important hyper-parameters are the number of local maxima $u$ and the number of thresholds $v$ which together control the number of proposals $p$ per image returned by the process. We study their influence on the colocalization performance by conducting experiments on VOC_6x2 and report the results in Table 3.1 (right). It shows that the colocalization performance does not depend much on the values of these parameters. Using $(u = 50, v = 100)$ actually gives the best performance but with twice as many proposals as $(u = 20, v = 50)$. For efficiency, we use $u = 20$ and $v = 50$ in all of our experiments.

We report in Table 3.2 the performance of OSD and rOSD on OD, VOC_6x2 and VOC_all with different types of proposals. It can be seen that our proposals give the best results on all datasets among all types of proposals with significant margins: 6.1%, 2.1% and 3.0% in colocalization and 5.3%, 0.5% and 4.7% in discovery, respectively. It is also noticeable that our proposals not only fare much better than the unsupervised ones (selective search [Uijlings, 2013] and edgeboxes [Zitnick, 2014]) but also outperform those generated by randomized Prim [Manen, 2013], an algorithm trained with bounding box annotation.

| Region proposals | Colocalization | | | Discovery | | |
|---|---|---|---|---|---|---|
| | OD | VOC_6x2 | VOC_all | OD | VOC_6x2 | VOC_all |
| [Zitnick, 2014] | 81.6 ± 0.3 | 54.2 ± 0.3 | 29.7 ± 0.1 | 81.4 ± 0.3 | 55.2 ± 0.3 | 32.6 ± 0.1 |
| [Uijlings, 2013] | 82.2 ± 0.2 | 54.5 ± 0.3 | 30.9 ± 0.1 | 81.3 ± 0.3 | 57.8 ± 0.2 | 33.0 ± 0.1 |
| [Manen, 2013] | 82.9 ± 0.3 | 71.5 ± 0.3 | 42.8 ± 0.1 | 82.5 ± 0.1 | 70.6 ± 0.4 | 44.5 ± 0.1 |
| Ours (OSD) | **89.0 ± 0.6** | **73.6 ± 0.6** | 44.7 ± 0.3 | **87.8 ± 0.4** | 69.2 ± 0.5 | 48.7 ± 0.3 |
| Ours (rOSD) | **89.0 ± 0.5** | 73.3 ± 0.5 | **45.8 ± 0.3** | 87.6 ± 0.3 | **71.1 ± 0.8** | **49.2 ± 0.2** |

Table 3.2 – Single-object colocalization and discovery performance of OSD with different types of proposals. We use VGG16 features [Simonyan, 2015a] to represent regions in these experiments. Best results are in bold, second best results are underlined.

We compare OSD and rOSD using our region proposals to the state of the art in Table 3.3 (colocalization) and 3.4 (discovery). In these experiments, we use VGG19 features [Simonyan, 2015a] to facilitate comparisons to [Li, 2016] and [Wei, 2019]. It can be seen that our use of CNN features (for both creating proposals and representing them in OSD) consistently improves the performance compared to the original OSD. It is also noticeable that rOSD performs significantly better than OSD on the two large datasets (VOC_all and VOC12) while on the two smaller ones (OD and VOC_6x2), their performances are comparable. It is due to the fact that images in OD and VOC_6x2 mostly contain only one well-positioned object thus bad local maxima are not a big problem in the optimization while images in VOC_all and VOC12 contain much

more complex scenes and the optimization works better with more regularization. In overall, we obtain the best results on the two smaller datasets, fare better than [Li, 2016] but are behind [Wei, 2019] on VOC_all and VOC12 in the colocalization setting. It should be noticed that while methods for image colocalization [Li, 2016; Wei, 2019] suppose that images in the collection come from the same category and explicitly exploit this assumption, rOSD is intended to deal with the much more difficult and general object discovery task. Indeed, in the discovery task, rOSD outperforms [Wei, 2019] by a large margin, 5.9% and 4.9% respectively on VOC_all and VOC12.

| Method | Features | OD | VOC_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| [Cho, 2015] | WHO | 84.2 | 67.6 | 37.6 | - |
| OSD† | WHO | 87.1 ± 0.5 | 71.2 ± 0.6 | 39.5 ± 0.1 | - |
| [Li, 2016] | VGG19 | - | - | 41.9 | 45.6 |
| [Wei, 2019] | VGG19 | 87.9 | 67.7 | **48.7** | **51.1** |
| Ours (OSD) | VGG19 | **90.3 ± 0.3** | 75.3 ± 0.7 | 45.6 ± 0.3 | 47.8 ± 0.2 |
| Ours (rOSD) | VGG19 | 90.2 ± 0.3 | **76.1 ± 0.7** | 46.7 ± 0.2 | 49.2 ± 0.1 |

Table 3.3 – Single-object colocalization performance of our approach compared to the state of the art. Note that [Wei, 2019] outperforms our method on VOC_all and VOC12 in this case, but the situation is clearly reversed in the much more difficult discovery setting, as demonstrated in Table 3.4. OSD† denotes the original OSD in Chapter 2.

| Method | Features | OD | VOC_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| [Cho, 2015] | WHO | 82.2 | 55.9 | 37.6 | - |
| OSD† | WHO | 82.3 ± 0.3 | 62.5 ± 0.6 | 40.7 ± 0.2 | - |
| [Wei, 2019] | VGG19 | 75.0 | 54.0 | 43.4 | 46.3 |
| Ours (OSD) | VGG19 | 89.1 ± 0.4 | 71.9 ± 0.7 | 47.9 ± 0.3 | 49.2 ± 0.2 |
| Ours (rOSD) | VGG19 | **89.2 ± 0.4** | **72.5 ± 0.5** | **49.3 ± 0.2** | **51.2 ± 0.2** |

Table 3.4 – Single-object discovery performance on the datasets with our proposals compared to the state of the art. OSD† denotes the original OSD in Chapter 2.

**Multi-Object Colocalization and Discovery.** We demonstrate the effectiveness of rOSD in multi-object colocalization and discovery on VOC_all and VOC12 datasets, which contain images with multiple objects. We compare the performance of OSD and rOSD to [Wei, 2019] in Table 3.5. Although the latter tackles only the single-object colocalization problem, we modify their method to have a reasonable baseline for the multi-object colocalization and discovery problem. Concretely, we take the bounding boxes around the 5 largest connected components of positive locations in the image's *indicator matrix* [Wei, 2019] as the localization results. It can be seen that our method obtains the best performance with significant margins to the closest competitor across all datasets and settings. It is also noticeable that rOSD, again, significantly outperforms OSD in this task. An illustration of multi-object discovery is shown in Figure 3.7. For a fair comparison, we use high values of $\nu$ (50) and $IoU$ (0.7) in the multi-object experiments to make sure that both OSD and rOSD return approximately 5 regions per image.

Images may of course contain fewer than 5 objects. In such cases, OSD and rOSD usually return overlapping boxes around the actual objects. We can often eliminate these overlapping boxes and obtain better qualitative results by using smaller $\nu$ and $IoU$ threshold values. It can be seen in Figure 3.7 that with $\nu = 25$ and $IoU = 0.3$, rOSD is able to return bounding boxes around objects without many overlapping regions. Note however that the quantitative results may worsen due to the reduced number of regions returned and the fact that many images contain objects that highly overlap, e.g., the last two columns of Figure 3.7. In such cases, a small $IoU$ threshold prevents discovering all of these objects.

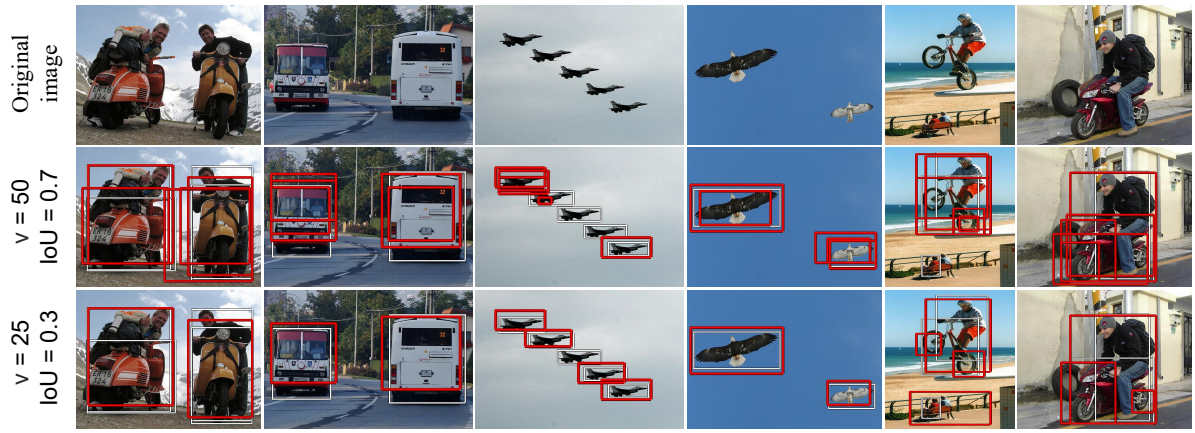



Figure 3.7 – Qualitative multi-object discovery results obtained with rOSD. White boxes are ground-truth objects and red ones are our predictions. Original images are in the first row. Results with $\nu = 50$ and $IoU = 0.7$ are in the second row. Results with $\nu = 25$ and $IoU = 0.3$ are in the third row.

| Method | Features | Colocalization | | Discovery | |
|---|---|---|---|---|---|
| | | VOC_all | VOC12 | VOC_all | VOC12 |
| OSD† | WHO | 40.7 ± 0.1 | - | 30.7 ± 0.1 | - |
| Wei *et al.* [Wei, 2019] | VGG19 | 43.3 | 45.5 | 28.1 | 30.3 |
| Ours (OSD) | VGG19 | 46.8 ± 0.1 | 47.9 ± 0.0 | 34.8 ± 0.0 | 36.8 ± 0.0 |
| Ours (rOSD) | VGG19 | **49.4 ± 0.1** | **51.5 ± 0.1** | **37.6 ± 0.1** | **40.4 ± 0.1** |

Table 3.5 – Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets. OSD† denotes the original OSD in Chapter 2.

**Large-Scale Object Discovery.** We apply our large-scale algorithm in the discovery task on VOC_all, VOC12 and C20K which are randomly partitioned respectively into 5, 10 and 20 parts of roughly equal sizes. In the first stage of all experiments, we prefilter the initial neighborhood of images and keep only 50 potential neighbors. We choose $\nu = 50$ and keep $K_1$ (which are 250, 500 and 1000 respectively on VOC_all, VOC12 and C20K) positive entries in each score matrix. In the second stage, we run rOSD (OSD) on the entire datasets with $\nu = 5$, limit the number of potential neighbors to 50 and use score matrices with only 50 positive entries. We choose $K_1$ such that each run in the first stage and the OSD run in the second stage have the

same memory cost, hence the values of $K_1$ chosen above. As baselines, we have applied rOSD (OSD) directly to the datasets, keeping 50 positive entries (baseline 1) and 1000 positive entries (baseline 2) in score matrices. Table 3.6 shows the object discovery performance on VOC_all, VOC12 and C20K for our large-scale algorithm compared to the baselines. It can be seen that our large-scale two-stage rOSD algorithm yields significant performance gains over the baseline 1, obtains an improvement of 6.6%, 9.3% and 4.0% in single-object discovery and 2.9%, 4.0% and 0.4% in multi-object discovery, respectively on VOC_all, VOC12 and C20K. Interestingly, large-scale rOSD also outperforms the baseline 2, which has a much higher memory cost, on VOC_all and VOC12.

| Method | Single-object | | | Multi-object | | |
|---|---|---|---|---|---|---|
| | VOC_all | VOC12 | C20K | VOC_all | VOC12 | C20K |
| Baseline 1 (OSD) | 41.1 ± 0.3 | 40.5 ± 0.2 | 43.6 ± 0.2 | 31.4 ± 0.1 | 32.4 ± 0.0 | 10.5 ± 0.0 |
| Baseline 1 (rOSD) | 42.8 ± 0.3 | 42.6 ± 0.2 | 44.5 ± 0.1 | 35.4 ± 0.2 | 37.2 ± 0.1 | 11.6 ± 0.0 |
| Baseline 2 (OSD) | 47.9 ± 0.3 | 49.2 ± 0.2 | - | 34.8 ± 0.0 | 36.8 ± 0.0 | - |
| Baseline 2 (rOSD) | 49.3 ± 0.2 | 51.2 ± 0.2 | - | 37.6 ± 0.1 | 40.4 ± 0.1 | - |
| Large-scale OSD | 45.5 ± 0.3 | 46.3 ± 0.2 | 46.9 ± 0.1 | 34.6 ± 0.0 | 36.9 ± 0.0 | 11.1 ± 0.0 |
| Large-scale rOSD | **49.4 ± 0.1** | **51.9 ± 0.1** | **48.5 ± 0.1** | **38.3 ± 0.0** | **41.2 ± 0.1** | **12.0 ± 0.0** |

Table 3.6 – Performance of our large-scale algorithm compared to the baselines. Our method and baseline 1 have the same memory cost, which is much smaller than the cost of baseline 2 . Also, due to memory limits, we cannot run baseline 2 on C20K.

**Results with the ensemble method from OSD** In Chapter 2, we use an ensemble method (EM) to combine several solutions before post processing to stabilize and improve the final performance of OSD. We investigate the influence of this procedure on the performance of OSD and rOSD with our new proposals, and present the result in Tables 3.7 and 3.8. We use VGG16 features in these experiments. It can be seen that the effect of EM is mixed for the tested datasets. It generally harms the performance on VOC_all and VOC12 and improves the performance on VOC_6x2 while its effect on OD is unclear. We have therefore chosen to omit EM in our experiments.

| Method | | OD | VOC_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| Ours (OSD) | w/o EM | 89.0 ± 0.6 | 73.6 ± 0.6 | 44.7 ± 0.3 | 49.0 ± 0.2 |
| Ours (OSD) | w/ EM | 88.2 ± 0.2 | **75.3 ± 0.2** | 44.7 ± 0.1 | 48.7 ± 0.1 |
| Ours (rOSD) | w/o EM | 89.0 ± 0.5 | 73.3 ± 0.5 | **45.8 ± 0.3** | **49.7 ± 0.1** |
| Ours (rOSD) | w/ EM | **89.2 ± 0.3** | 74.5 ± 0.2 | 45.5 ± 0.1 | **49.7 ± 0.2** |

Table 3.7 – Influence of the ensemble method on the colocalization performance of OSD and rOSD with our proposals.

**Evaluating the graph computed by rOSD** Following [Cho, 2015], we evaluate the local graph structure obtained by rOSD using the CorRet measure ('Av.' version). Given the image

| Method | | OD | VOC_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| Ours (OSD) | w/o EM | 87.8 ± 0.4 | 69.2 ± 0.5 | 48.7 ± 0.3 | 51.3 ± 0.2 |
| Ours (OSD) | w/ EM | 87.5 ± 0.3 | 70.9 ± 0.3 | 48.6 ± 0.1 | 50.7 ± 0.1 |
| Ours (rOSD) | w/o EM | 87.6 ± 0.3 | 71.1 ± 0.8 | **49.2 ± 0.2** | **52.1 ± 0.1** |
| Ours (rOSD) | w/ EM | **88.7 ± 0.3** | **71.9 ± 0.4** | 48.7 ± 0.1 | 52.0 ± 0.1 |

Table 3.8 – Influence of the ensemble method on the single-object discovery performance of OSD and rOSD with our proposals.

neighbors found for images in a class, the class CorRet is defined as the average percentage of the image neighbors that also belong to that class. The final CorRet is the average class CorRet over all classes. As a baseline, we consider the local graph induced by the sets of nearest neighbors $N(i)$ computed from the fully connected layer *fc6* of the CNN that are used in the same experiment. Table 3.9 shows the CorRet of local graphs obtained when running rOSD (OSD) on VOC_all and VOC12 and large-scale rOSD (OSD) on C20K in the mixed setting. It can be seen that the local image graph returned by our methods has significantly higher CorRet than the baseline.

| Dataset | VOC_all | VOC12 | C20K |
|---|---|---|---|
| Baseline | 50.7 | 56.4 | 36.8 |
| Ours (OSD) | **60.1 ± 0.1** | **63.2 ± 0.0** | **39.8 ± 0.0** |
| Ours (rOSD) | 59.8 ± 0.1 | 63.0 ± 0.0 | 39.4 ± 0.0 |

Table 3.9 – Quality of the returned local image graph as measured by CorRet.

**Execution time.** Similar to the original OSD, rOSD requires computing the similarity scores for a large number of image pairs which makes it computationally costly. It takes in total 478 paralellizable CPU hours, 300 unparallelizable CPU seconds and 1 GPU hour to run single-object discovery on VOC_all with 3550 images. This is more costly compared to only 812 GPU seconds needed by [Wei, 2019] but is less costly than the original OSD using CNN features. The latter requires 546 paralellizable CPU hours, 250 unparalellizable CPU seconds and 4 GPU hours. Note that the unparallelizable computational cost, which comes from the main OSD algorithm, grows very fast (at least linearly in theory, it takes 2.3 hours on C20K in practice) with the data set's size and is the time bottleneck in large scale.

## 3.5 Conclusion and Limitations

We have addressed in this chapter the limitations of OSD. We have presented an unsupervised algorithm for generating region proposals from CNN features trained on an auxiliary and unrelated task. Our proposals come with an intrinsic structure which can be leveraged as an additional regularization in the OSD framework. The combination of our new proposals and regularized OSD gives comparable results to the current state of the art in image colocalization,

set at the time of the writing of our article [Vo, 2020] a new state-of-the-art single-object discovery and has proven effective in the multi-object discovery. We have also successfully extended OSD to the large-scale case and show that our method yields significantly better performance than plain OSD.

Although we have scaled unsupervised object discovery to datasets five times larger than those considered in Chapter 2, scaling it further to datasets several orders of magnitude larger is not trivial since the second stage of our large-scale algorithm still has to run on the entire image collection. Moreover, while the second loop, the ascent of $e$, in Algorithm 3.1 is parallelizable, the first loop, the ascent of $x$, is inherently sequential. Indeed, the update of $x_i$ depends on the values of other $x_j$ with $j \neq i$ and a parallel execution of these updates would not guarantee the improvement of the objective. Running with multiple machines in a distributed manner, a common solution to overcome the scale issue, is therefore not applicable. Besides, using a reduced set of region proposals in the second stage of the large-scale rOSD algorithm could potentially hurt the model's ability to discover many objects per image. In the next chapter, we will introduce a new formulation to overcome these issues, enabling effective unspervised object discovery on very large datasets.

# Chapter 4

# Large-Scale Unsupervised Object Discovery

Objectives

Existing approaches to unsupervised object discovery do not scale up to large datasets without approximations that compromise their performance. We propose a novel formulation of UOD as a ranking problem, amenable to the arsenal of distributed methods available for eigenvalue problems and link analysis. Through the use of self-supervised features, we also demonstrate the first effective fully unsupervised pipeline for UOD. Extensive experiments on COCO and OpenImages datasets show that, in the single-object discovery setting where a single prominent object is sought in each image, the proposed LOD (Large-scale Object Discovery) approach is on par with, or better than the previous state of the art for medium-scale datasets (up to 120K images), and over 37% better than the only other algorithms capable of scaling up to 1.7 M images. In the multi-object discovery setting where multiple objects are sought in each image, the proposed LOD is over 14% better in object discovery Average Precision (odAP) than all other methods for datasets ranging from 20K to 1.7M images. Using self-supervised features, we also show that the proposed method obtains state-of-the-art UOD performance on OpenImages.
This work, done in collaboration with Elena Sizikova, Cordelia Schmid, Patrick Pérez and Jean Ponce, has appeared in Proceedings of the Conference on Neural Information Processing Systems (NeurIPS) 2021.

## Contents


4.1 Introduction . . . . . **57**
4.2 Problem Statement and Related Work . . . . . **59**
4.2.1 Problem Statement . . . . . 59
4.2.2 Related Work . . . . . 60
4.3 Proposed Approach . . . . . **60**
4.3.1 Quadratic Formulation . . . . . 60
4.3.2 PageRank Formulation . . . . . 62
4.3.3 Using (Q) to Personalize PageRank . . . . . 62
4.4 Experimental Analysis . . . . . **63**
4.4.1 Large-scale Object Discovery . . . . . 65
4.4.2 Category Discovery . . . . . 70
4.4.3 Discussions . . . . . 73
4.5 Conclusion and Future Work . . . . . **74**



Figure 4.1 – Sample unsupervised object discovery results obtained by LOD on the OpenImages dataset [Krasin, 2017] which contains 1.7M images. Ground-truth boxes are shown in yellow, and predictions are in red. Best viewed in color.

## 4.1 Introduction

It is natural to cast unsupervised object discovery as the task of finding repetitive visual patterns in image collections. In the previous chapters, we have formulated it as a combinatorial optimization problem in an image graph, selecting simultaneously image pairs that contain similar objects and region proposals that correspond to objects. This formulation, however, comes with several computational limitations which hinder its application to very large datasets. The motivation behind this chapter is to formulate UOD as a simpler graph-theoretical problem with a more efficient solution, where objects correspond to well-connected nodes in a graph whose nodes are region proposals (instead of images in OSD and rOSD), and edges are weighted by region similarity and "objectness". In this scenario, identifying the most promising object-proposal nodes is a ranking problem where the goal is to rank the nodes based on how well they are connected in the graph. From another perspective, ranking is rather a natural modeling choice for UOD since, in our context, discovering objects means finding the most object-like regions in a set of initial region proposals, which naturally amounts to ranking them according to their objectness. As a result, a large array of methods available for eigenvalue problems [Landau, 1895] and link analysis [Page, 1999] can be applied to solve UOD on much larger datasets than previously possible (Figure 4.1). The proposed pipeline for unsupervised object discovery is illustrated in Figure 4.2.

We consider three variants of this approach: the first one re-defines the UOD objective of OSD and rOSD as an eigenvalue problem on the graph of region proposals, the second variant explores the applicability of PageRank [Brin, 1998; Page, 1999] to UOD, and the final one combines the other two into a hybrid algorithm, dubbed LOD (for Large-scale Object Discovery), which uses the solution of the eigenvalue problem to personalize PageRank. LOD offers a fast, distributed solution to object discovery on very large datasets. We show in Section 4.4.1 and Table 4.1 that its performance is comparable or better than the state of the art in the single object discovery setting for datasets of up to 120K images, and over 37% better than the only algorithms we are aware of that can handle up to 1.7M images. In the multi-object discovery setting, LOD significantly outperforms all existing techniques on datasets containing from 20K to 1.7M images. While LOD does not explicitly address discovering relationships between images (e.g., grouping them into classes), we demonstrate that categories can be discovered in a post-processing step (see Section 4.4.2). The best performing approaches to UOD at the time of publication all use

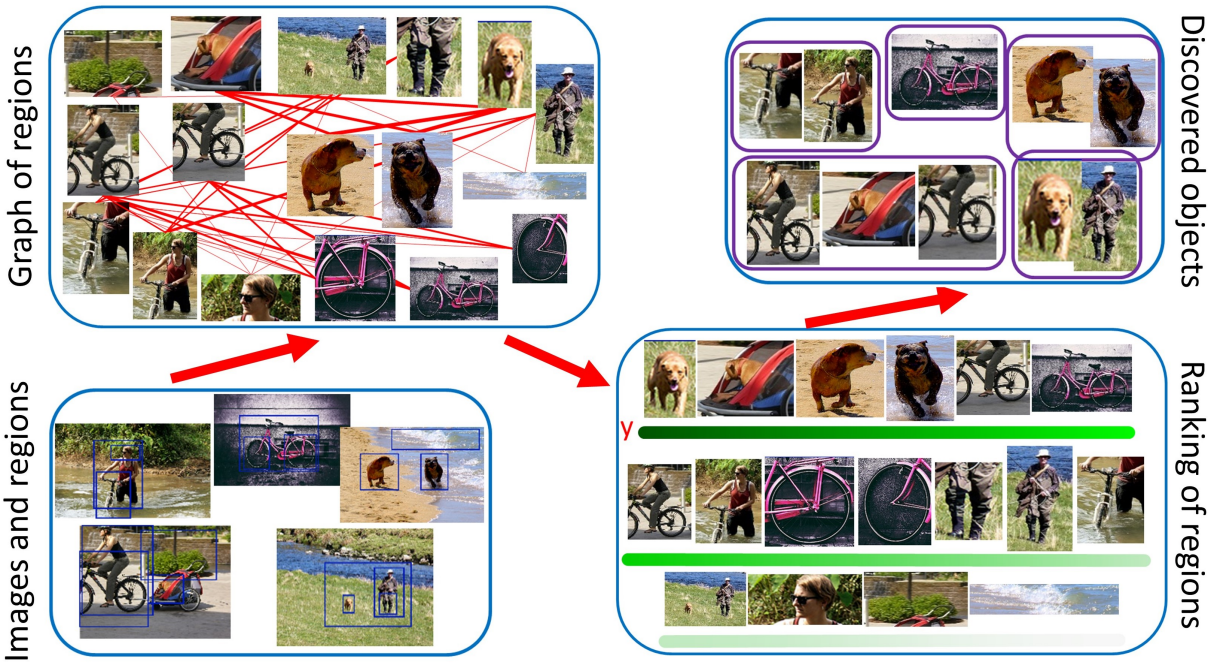



Figure 4.2 – An overview of the proposed method LOD. It receives as input a collection of images, each equiped with a set of region proposals. It then builds an undirected weighted graph where nodes are regions and edge weights reflect node similarity. Ranking methods are then applied to obtain for each node-region a score. Finally, the regions with highest scores in each image are returned as objects.

*supervised* region proposals [OSD] and/or features [Wei, 2019; rOSD]. We also demonstrate for the first time in Section 4.4.1 that self-supervised features can give good UOD performance. Our main contributions can be summarized as follows:

- We propose a new formulation of UOD as a ranking problem, allowing the application of parallel and distributed link analysis methods [Landau, 1895; Brin, 1998; Page, 1999].
- We scale UOD up to datasets 87 times larger than those considered in rOSD. Our novel LOD algorithm outperforms others on medium-size datasets by up to 32%.
- We propose to use self-supervised features for UOD and show that LOD, combined with these features, offers a viable UOD pipeline without any supervision whatsoever.
- We conduct extensive experiments on the COCO [Lin, 2014] and OpenImages [Krasin, 2017] datasets to empirically validate our method. We also demonstrate applications of our approach to object category discovery and image neighbor retrieval, outperforming other existing unsupervised baselines on both tasks by a large margin.

| Method | Single-object | | | | Multi-object | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CorLoc | | | | odAP50 | | | | odAP[50:95] | | | |
| | C20K | C120K | Op50K | Op1.7M | C20K | C120K | Op50K | Op1.7M | C20K | C120K | Op50K | Op1.7M |
| [Zitnick, 2014] | 28.8 | 29.1 | 32.7 | 32.8 | 4.86 | 4.91 | 5.46 | 5.49 | 1.41 | 1.43 | 1.53 | 1.53 |
| [Wei, 2019] | 38.2 | 38.3 | 34.8 | 34.8 | 2.41 | 2.44 | 1.86 | 1.86 | 0.73 | 0.74 | 0.6 | 0.6 |
| [Kim, 2009] | 35.1 | 34.8 | 37.0 | - | 3.93 | 3.93 | 4.13 | - | 0.96 | 0.96 | 0.98 | - |
| rOSD | **48.5** | 48.5 | 48.0 | 47.8 | 5.18 | 5.03 | 4.98 | 4.88 | 1.62 | 1.6 | 1.58 | 1.57 |
| LOD+Self [Gidaris, 2021] | 41.1 | 42.4 | **49.5** | **49.4** | 4.56 | 4.90 | 6.37 | **6.28** | 1.29 | 1.37 | 1.87 | **1.86** |
| LOD | **48.5** | **48.6** | 48.1 | 47.7 | **6.63** | **6.64** | **6.46** | **6.28** | **1.98** | **2.0** | **1.88** | 1.83 |

Table 4.1 – Large-scale object discovery performance and comparison to the state of the art on COCO [Lin, 2014] (C120K), OpenImages [Krasin, 2017] (Op1.7M) and their respective subsets C20K and Op50K, in three standard metrics. Using VGG16 features [Simonyan, 2015a], the proposed method LOD achieves top performance in both single and multi-object discovery, and scales better to 1.7M images in Op1.7M than rOSD. When running with self-supervised features (LOD + Self [Gidaris, 2021]), it yields the best results on Op1.7M, showing the first effective fully unsupervised pipeline for UOD. See Section 4.4 for more details.

## 4.2 Problem Statement and Related Work

### 4.2.1 Problem Statement

The object discovery problem are formulated in the previous chapters as the following combinatorial maximization problem:

$$\max_{x,e} \sum_{p=1}^{n} \sum_{q \in \mathcal{N}(p)} e_{pq} x_p^T S_{pq} x_q \quad \text{s.t.} \sum_{k=1}^{r} x_p^k \leq \nu \text{ and } \sum_{q \neq p} e_{pq} \leq \tau \quad \forall\ 1 \leq p \leq n, \tag{C}$$

where $S_{pq} \in \mathbb{R}^{r \times r}$ is a matrix whose entry $S_{pq}^{k\ell} \geq 0$ measures the similarity between region $k$ of image $p$ and region $\ell$ of image $q$ as well as the saliency of the respective regions, $\mathcal{N}(p)$ is a set of potential high-similarity neighbors of image $p$, and $\nu$ and $\tau$ are predefined constants corresponding to the maximum number of objects in an image and the maximum number of its neighbors, respectively. For the sake of simplicity, we assume in this chapter that all images have exactly $r$ region proposals. OSD and rOSD solve a convex relaxation of (C) in the dual domain and/or use block-coordinate ascent on its variables $x$ and $e$. The similarity scores $S_{pq}^{k\ell}$ are typically computed using the Probabilistic Hough Matching (PHM) algorithm from [Cho, 2015], which combines local appearance and global geometric consistency constraints to compare pairs of regions. A high PHM score between a pair of proposals is an indicator of whether the corresponding two proposals may correspond to a common foreground object. We follow this tradition in LOD and also use PHM scores (Section 4.4).

The objective of UOD as formulated in (C) is to find both the objects (variables $x_p^k$) and the edges linking the images that contain them (variables $e_{pq}$). Its combinatorial nature makes it hard to scale up to large values of $n$ and $r$. rOSD uses a block-coordinate ascent algorithm to (C), updating variables $x$ and $e$ alternatively to optimize the objective (Algorithm 3.1). It attempts to scale up (C) with a drastic approximation, running on parts of the image collection to reduce $r$ to only 50 before running on the entire dataset. However, using significantly reduced

sets of region proposals hinders its ability to discover multiple objects (Table 4.1). Moreover, the sequential nature of this algorithm (Section 3.5) prevents it from scaling up to datasets with millions of images. We therefore drop the second objective of UOD, and rely only on a fully connected, weighted graph of proposals where edge weights encode proposal similarity (edge weights can be zeros, see Section 4.3). In turn, we can reformulate UOD as a ranking problem [Landau, 1895; Katz, 1953; Pinski, 1976; Brin, 1998; Kleinberg, 1999], amenable to the panoply of large-scale distributed tools available for eigenvalue problems and link analysis. We consider two different ranking formulations: the first one (Q) tackles a quadratic optimization problem, and the second (P) is based on the well-known PageRank algorithm [Brin, 1998; Page, 1999]. We combine these two approaches into a joint formulation (LOD) that gives the best results on large-scale datasets. See Sections 4.3 and 4.4 for details.

### 4.2.2 Related Work

**Applications of ranking to computer vision.** The goal of ranking is to assign a global importance rating to each item in a set according to some criterion [Page, 1999]. Many computer vision problems admit a ranking formulation, including image retrieval [Cakir, 2019], object tracking [Bai, 2012], person re-identification [Loy, 2013], video summarization [Yao, 2016], co-segmentation [Quan, 2016b] and saliency detection [Li, 2015; Siméoni, 2019]. Several techniques specifically designed for large-scale ranking problems [Kleinberg, 1999; Page, 1999] have been used to explore large datasets of images [Jing, 2008] and shapes [Funkhouser, 2003]. PageRank-based approaches in particular have been popular [Jing, 2008; Ren, 2018; Payandeh, 2019] due to their scalability. [Kim, 2008] proposes an algorithm for object discovery that combines appearance and geometric consistency with PageRank-based link analysis for category discovery. However, it does not scale beyond 600 images. A more scalable follow-up work by [Kim, 2009] discovers regions of interest (RoIs) from images with successive applications of PageRank. This algorithm includes two main steps. The first one attempts to find object representatives (*hubs*) from the current RoIs of all images using PageRank. PageRank is then utilized again in the second step to analyze the links between regions in each image and the hubs, this time to update the RoIs of the images. Finally, good RoIs are found by repeating these two steps until convergence. We compare our method to this technique in Section 4.4.1.

## 4.3 Proposed Approach

### 4.3.1 Quadratic Formulation

Let us represent region proposals by a graph $G$ with $N = nr$ nodes, where $n$ is the number of images and $r$ is the number of proposals in each image. Each node $(p, k)$ corresponds to proposal $k$ of image $p$, and any two nodes $(p, k)$ and $(q, \ell)$ are linked by an edge with weight $S_{pq}^{k\ell}$. Graph $G$ is represented by an $N \times N$ symmetric adjacency matrix $W$, consisting of $r \times r$ blocks $S_{pq}$ for $p, q = 1, \dots n$. $S_{pq}$ is defined in Section 4.2.1 and is computed by the PHM algorithm [Cho, 2015] if $p \neq q$, and the diagonal blocks are taken to be zero since only inter-image region similarity

matters in our setting. Let $y_i \geq 0$ denote some measure of importance that we want to estimate for node $i$ and let $y = (y_1, \ldots, y_N)^T$, we define the *support* of node $i$ given $y$ as $z_y(i) = \sum_j W_{ij} y_j$ so that, taking $z_y = (z_y(1), \ldots, z_y(N))^T$, we have $z_y = Wy$. Intuitively, given $y$, $z_y(i)$ quantifies how well $i$ is connected to (or "supported by") the rest of the nodes $j$ in the graph, taking into account the similarity $W_{ij}$ between $i$ and $j$ as well as the importance $y_j$ of that node. We would like to find the importance scores that *rank* the nodes as well as possible, so that the order corresponds to their amount of support. As shown by the following lemma, it turns out that this "chicken-and-egg" problem admits a simple solution.

**Lemma 4.3.1.** *Suppose $W$ is irreducible (i.e., represents a strongly connected graph $G$). The solution $y^*$ of the quadratic optimization problem:*

$$y^* = \underset{\|t\| \leq 1, t \geq 0}{\arg\max}\, t^T W t \tag{Q}$$

*is the unique unit, non-negative eigenvector of $W$ associated with its largest eigenvalue.*

This is a classic result and can be proved using the Perron-Frobenius theorem [Frobenius, 1912; Perron, 1907]. We include the complete proof in Appendix C. In our context, $W$ is not, in general, irreducible since some proposal similarities may be zero. Reminiscent of PageRank [Page, 1999], we add a small term $\frac{1}{N}\gamma e e^T$ to $W$, with $e$ being the vector with all entries equal to 1 in $\mathbb{R}^N$ and $\gamma = 10^{-4}$, deliberately chosen small so that the added term does not influence the similarity score much, to make $W$ irreducible. This term ensures that the resulting ranking is unique and serves the same purpose as the similar term in PageRank.

**Note:** since $y^*$ is associated with $W$'s largest eigenvalue $\lambda^*$, which is positive according to the Perron-Frobenius theorem, we have $\lambda^* y^* = W y^* = z_{y^*}$. Hence, the importance score $y_i^*$ of each node is, up to a positive constant, equal to its support, and can thus be used to rank the nodes as desired. Notice that (C) and (Q) are closely related problems when the graph of images in (C) is assumed to be complete, i.e., all images are connected in the graph. In this case, (C) can be written as $\max_{x \in \{0,1\}^N} x^T W x$, s.t., for all $p$ from 1 to $n$, $\sum_{k=1}^{r} x_{r(p-1)+k} \leq \nu$. Here, we stack $x_i$ $(i = 1, \ldots, n)$ into a vector $x$. (Q) can thus be seen as a continuous relaxation of (C) where the binary variables are replaced by continuous ones, and the linear constraints attaching the proposals to their source images are dropped. The order induced by the dominant eigenvector $y^*$ of $W$ on the nodes of $G$ is reminiscent of the PageRank approach [Brin, 1998; Page, 1999] to link analysis. This remark leads to a second approach to UOD through ranking, discussed next.

### 4.3.2 PageRank Formulation

When defining PageRank, [Page, 1999] does not start from an optimization problem like (Q), but directly formulates ranking as an eigenvalue problem. Following [Langville, 2004], let $A$ denote the transition matrix of the graph associated with a Markov chain, such that $a_{ij} \geq 0$ is the probability of moving from node $j$ to node $i$. In our context, $A$ can be taken as $WD^{-1}$ where $D$ is the diagonal matrix with $D_{jj} = \sum_i W_{ij}$. By definition [Brin, 1998; Page, 1999], the

PageRank vector $v$ associated with the matrix $A$ is the unique non-negative eigenvector $v$ of the matrix $P$, associated with its largest (unit) eigenvalue, where $P$ is defined as:

$$P = (1 - \beta)A + \beta u e^T, \tag{P}$$

with $\beta$ is a damping factor. Here, $u$, the so-called personalized vector, is an element of $\mathbb{R}^N$ such that $e^T u = 1$. As noted earlier, the second term ensures that $P$ is irreducible, so that, by the Perron-Frobenius theorem, the eigenvector $v \geq 0$ is unique [Langville, 2011]. The vector $u$ is typically taken equal to $\frac{1}{N}e$, but can also be used to "personalize" the ranking by attaching more importance to certain nodes. This leads to the hybrid formulation proposed in the next section. (Q) and (P) are closely related, and the vector $v$ can also be seen as the solution of a quadratic optimization problem [Mahoney, 2010]. Besides this formal similarity, the goals of the two formulations are also similar. Quoting [Page, 1999], "a page has a high rank (according to PageRank) if the sum of the ranks of its backlinks is high". The solution of both (Q) and (P), as an eigenvector associated with the largest eigenvalue, provides a ranking based on the support function and can be found with the power iteration algorithm [Mises, 1929]. This algorithm involves only matrix-vector multiplications and can be implemented efficiently in a distributed way.

### 4.3.3 Using (Q) to Personalize PageRank

The above discussion suggests combining the two approaches. We thus propose to use the maximizer of (Q) to generate the personalized vector for (P). (Q) and (P) are two different optimization problems for ranking region proposals, and combining them may help improve the final performance. Intuitively, region proposals with high scores given by (Q) are reliable and we should be able to rank the "objectness" of other regions more accurately based on the "feedback" of these top-scoring proposals. We compute the personalized vector from the solution of (Q) as follows. Given a factor $\alpha$, the top region in each image are chosen as candidates, then the top $\alpha$ percent of regions amongst these candidates are selected. Since only regions that have a high probability of being correct are beneficial, we choose $\alpha$ sufficiently small (see Table 4.4) to select only the regions most likely to be correct. Given the set of selected regions, the personalized vector $u$ is the $L_1$-normalized indicator vector with $u_i = 1/K$ if proposal $i$ is selected and $u_i = 0$ otherwise, where $K$ is the total number of selected regions. We set the initialization $v_0$ of the power iteration algorithm (see Algorithm 4.1) to $u$ to further bias (P) toward reliable regions found by (Q). In what follows, we refer to this hybrid algorithm as Large-Scale Object Discovery (LOD).

## 4.4 Experimental Analysis

**Datasets.** We consider two large public datasets: C120K, a combination of all images in the training and validation sets of the COCO 2014 dataset [Lin, 2014], except those that contain only "crowd" objects, with approximately 120,000 images depicting 80 object classes and OpenImages (Op1.7M) [Krasin, 2017], the largest dataset ever evaluated for UOD so far, with 1.7 million

images. The latter dataset is 87 times the size of the previous largest dataset evaluated by rOSD. We resize all images in this dataset so that their largest side does not exceed 512 pixels. To facilitate ablation studies and comparisons, we also evaluate our methods on C20K, the subset of C120K containing 19,817 images used by rOSD and Op50K, a subset of Op1.7M containing 50,000 images.

**Implementation details.** We use the proposal generation method proposed in rOSD since it gives the best object discovery performance among the unsupervised region proposal extraction methods (see Chapter 3). We use VGG16 [Simonyan, 2015a], trained with and without image class labels (Section 4.4.1) on the ImageNet [Deng, 2009] dataset, to both generate (with the method in rOSD) and represent (extracting with RoiPool [Girshick, 2015]) proposals. We have also experimented with VGG19 [Simonyan, 2015a] and ResNet101 [He, 2016], but found they give worse performance, possibly because they are more discriminative and less helpful in localizing entire objects. We compute the similarity score between proposals with the PHM algorithm [Cho, 2015] similar to OSD and rOSD. For large datasets, computing all score matrices $S_{pq}$ is intractable. In this case, we only compute the similarity scores for the 100 nearest neighbors of each image, computed based on the Euclidean distance between image features from the *fc6* layer. For optimization, we choose $\beta = 10^{-4}$ in (P) and $\alpha = 10\%$ in LOD, and discuss LOD's sensitivity to these parameters in Table 4.4. To select objects from ranked proposals in an image, we choose proposal $i$ as an object if it has the highest score in the image or the intersection over union (IoU) between $i$ and each of the previously selected object regions is at most 0.3. When using proposals from rOSD, which are divided into disjoint groups, we additionally impose that the newly chosen region must be in a group different from the groups of the previously selected objects.

**Parallel power iterations.** We solve (Q), (P), and LOD with a parallel version of the power iteration method [Mises, 1929]. Since the adjacency matrix in (Q) and the PageRank matrix in (P) are very large, we divide them into chunks of consecutive rows of approximately equal size. At iteration $t$ in the optimization, these chunks are loaded in parallel into the memory of multiple processors for multiplication with the current iterate $x_t$. The results of these operations are chunks of the new vector $x_{t+1}$ which is then assembled from them in the main processor. We run up to $T = 50$ iterations of the power method in each experiment. The parallel power iteration algorithm is summarized in Algorithm 4.1.

**Metrics and evaluation settings.** Similar to rOSD, we consider two settings: single- and the multi-object discovery. In the single-object setting, we return $m = 1$ region per image, which is the region most likely to be an object. In the multi-object setting, we return up to $M$ regions per image, where $M$ is the maximum number of objects in any image in the dataset. Following [Locatello, 2020], we assume $M$ is known during evaluation. In a real application, one could use a rough "budget estimate" of the upper bound on how many objects per image one may try to detect. Measuring performance of UOD is always a difficult task due to the ambiguity of the

**Algorithm 4.1:** Parallel power iterations for finding the first eigenvector of a matrix $A$.

**Input:** Number $M$ of matrix chunks, chunks of rows $A_1,\ldots,A_M$ of $A$, number $N$ of rows of $A$, norm $L_p$ ($p=1$ for (P) and $p=2$ for (Q)), number $T$ of iterations.

**Result:** The first eigenvector of $A$.

$v_0 \leftarrow \frac{1}{\|e_N\|_p} e_N$ ▷ Initialize the iterate

**for** $t = 0$ *to* $T-1$ **do**

  In parallel in multiple processors, do

  **for** $i = 1$ *to* $M$ **do**

    Load matrix chunk $A_i$ into memory

    $v_{t+1,i} \leftarrow A_i v_t$ ▷ Compute the $i$-th chunk of the iterate

  **end**

  In the main processor, do

    $v_{t+1} \leftarrow [v_{t+1,1}; v_{t+1,2}; \ldots; v_{t+1,M}]$ ▷ Assemble the iterate from its chunks

  $v_{t+1} \leftarrow \frac{1}{\|v_{t+1}\|_p} v_{t+1}$ ▷ Normalize the iterate

**end**

Return $v_T$.

notion of an object in an unsupervised setting: object parts *vs.* objects, individual objects *vs.* crowd objects, *etc.* We follow the tradition of [Cho, 2015], OSD and rOSD and consider the annotated bounding boxes in the tested datasets as the only correct objects and use them to evaluate our methods. We evaluate UOD results according to two metrics, *Correct localization score* (CorLoc) and *object discovery Average Precision* (odAP). CorLoc is defined in the previous chapters but for convenience, we provide below the definitions of both metrics:

1. CorLoc – percentage of images correctly localized, i.e., where the IoU score between one of the ground-truth regions and the top predicted region is at least $\sigma = 0.5$. Note that it is equivalent to precision of returned regions. This metric is commonly used to evaluate single-object discovery.
2. odAP – the area under the precision-recall curve with precision and recall computed at each value of $m$ from 1 to $M$. A ground-truth object is considered discovered if its intersection with any predicted region is at least $\sigma$. This metric is used to evaluate multi-object discovery. We report odAP50 where $\sigma = 50$ and odAP[50:95], where we average odAP at 10 equally spaced values of $\sigma$ from 0.5 to 0.95. These two metrics are similar to AP50 and AP[50:95], the standard metrics for object detection [Gidaris, 2015; Girshick, 2015; Girshick, 2014; He, 2017; Redmon, 2016; Redmon, 2017; Ren, 2015a]. Note that odAP is different from the *detection rate* metric for multi-object discovery used in rOSD, which is the object recall at a predefined value of $m$. This metric depends on the number of selected regions per image $m$ while odAP does not. Also, since the precision decreases significantly with increasing $m$, odAP appears much smaller than CorLoc.

### 4.4.1 Large-scale Object Discovery

In this section, we compare our methods to the state of the art in unsupervised object discovery [Kim, 2009; Wei, 2019; rOSD]. We also compare to Edgeboxes [Zitnick, 2014], an unsupervised method which outputs regions with an importance score. Edgeboxes is a baseline of the type of information bounding boxes alone can provide in our setting. For a fair comparison, we have re-implemented [Kim, 2009] using supervised VGG16 features [Simonyan, 2015a] and proposals from rOSD. For [Wei, 2019], we modified the authors' public code, taking bounding boxes around more than one connected component of positive locations from the image *indicator matrix* to return more regions.

**Quantitative evaluation.** We evaluate baselines and the proposed method on C20K, C120K, Op50K and Op1.7M in Table 4.1. Since state-of-the-art approaches to UOD report results using supervised features [Simonyan, 2015a], we have used these features as well in our comparisons. We additionally report LOD's performance with self-supervised features [Gidaris, 2021] on these datasets. Overall, LOD obtains state-of-the-art object discovery performance in all settings and datasets. Using VGG16 features [Simonyan, 2015a], it outperforms [Kim, 2009], [Wei, 2019] and Edgeboxes [Zitnick, 2014] by large margins: 26% in single-object discovery and by 14% in multi-object discovery settings. In comparison to rOSD, LOD performs similarly in the single-object setting, but outperforms rOSD by at least 19% in the multi-object setting. This is likely due to the fact that our proposed LOD method considers the full proposal graph and does not reduce the number of region proposals (see Table 4.5). It is also noteworthy that LOD scales better than rOSD and runs much faster on the large datasets C120K and Op1.7M (Figure 4.3). On the Op1.7M dataset, it takes 53.7 hours to run while rOSD needs more than a month to finish. It is also interesting that self-supervised features [Gidaris, 2021] works better with LOD than supervised ones [Simonyan, 2015a], yielding the state-of-the-art performance on Op1.7M dataset.

**Run time.** Next, we compare scalability and run times of the proposed technique and of the baselines. All tested methods [Kim, 2009; Wei, 2019; Zitnick, 2014; rOSD; LOD] use similar pre-processing steps: Feature extraction, proposal generation and similarity computation, which are done separately across all images. This is followed in [Kim, 2009], rOSD and LOD by an optimization stage. The optimization step in rOSD is inherently sequential, but [Kim, 2009] and LOD can be parallelized. In our experiments, we use 4,000 CPUs for preprocessing for all methods, and 48 CPUs for the optimization step in [Kim, 2009] and LOD, the maximum possible with the MatLab parallel toolbox used in our implementation. The timings in Figure 4.3 include both pre-processing and optimization, when the latter is used. It can be seen that [Kim, 2009; Wei, 2019; Zitnick, 2014] and LOD scale nearly linearly with the number of images, while rOSD exhibits a superlinear pattern. Note that [Zitnick, 2014] and [Wei, 2019] are 70 times faster than LOD, but at a significant decrease in performance. These methods are not initially designed for object discovery, but serve as good, scalable baselines. Compared to previous top UOD methods, LOD runs at least 2.8 times faster than [Kim, 2009] on all datasets, at least 2 times

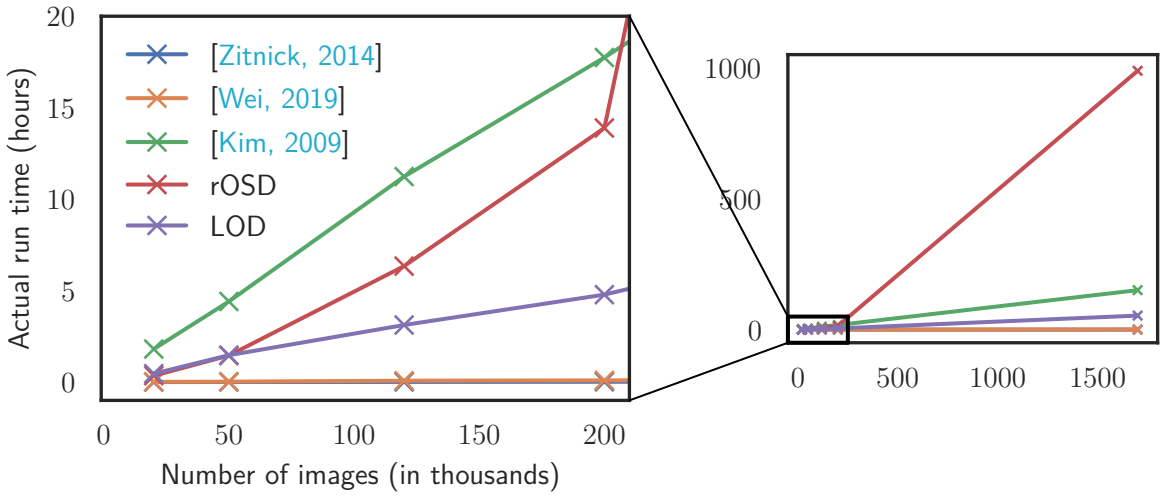



Figure 4.3 – Comparison of run time as a function of the number of input images. LOD achieves significant improvement in performance and/or savings in run time compared to previous works. [Zitnick, 2014] and [Wei, 2019] are linear in the number of images but their run time are very small compared to other methods and look flat in the figure.

faster than rOSD on datasets between 120K and 1.7 million images. Here, we evaluate only the parallel implementation typical for modern computing setups. In a serial implementation, compute times will be similar between top performing UOD methods [Kim, 2009], rOSD and LOD, but none of the methods would be able to run on 1.7M images in reasonable time. Note also that additional computational resources can further speed up processing for both [Kim, 2009] and LOD.

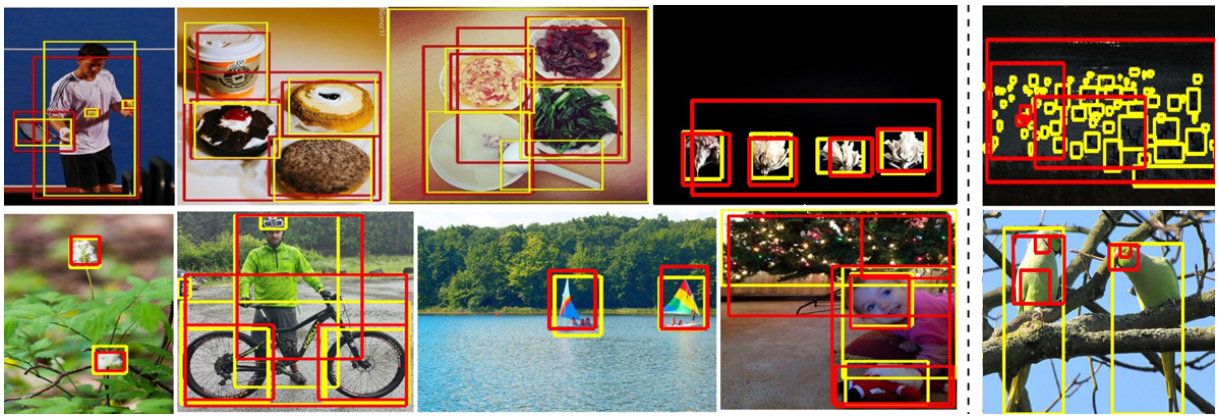


Figure 4.4 – Examples where our method (LOD) succeeds (left) and fails (right) to discover ground-truth objects in the Op1.7M dataset [Krasin, 2017]. Ground-truth objects are in yellow, our predictions are in red. Best viewed in color.

**Qualitative evaluation.** We present sample qualitative multi-object discovery results of LOD on C120K and Op1.7M in Figure 4.4 (additional Op1.7M results are presented in Figure 4.1). LOD discovers both the larger objects (people in the first and sixth images on the left, food

items in the second and third images) and the smaller ones (tennis balls and racket in the first image). It may fail of course, and two typical failure cases are shown on the right of Figure 4.4. In the first case, objects are too small and in the second case, LOD returns object parts instead of entire objects. Note that there is some ambiguity in what parts of the image are labelled as ground-truth objects. For example, the leaves in the bottom left image are not labelled as objects, while the flowers are.

**Self-supervised *vs.* supervised features.** LOD and all of the optimization-based baselines [Kim, 2009; Wei, 2019; rOSD] rely on a VGG [Simonyan, 2015a]-based classifier trained on ImageNet [Deng, 2009]. In this section, we investigate their performance when the underlying classifier is trained with ('Sup') and without ('Self') image labels, i.e., in a self-supervised fashion. To obtain self-supervised features, we use a VGG16 model trained with OBoW [Gidaris, 2021], a recent method which yields state-of-the-art performance in object detection after fine-tuning. This model is tested for both the proposal generation and similarity computation steps in optimization-based methods. The results of several variants of each optimization method, depending on the proposal generation algorithm ([Zitnick, 2014], rOSD+Self or rOSD+Sup) and the region proposal representation (Self or Sup) are presented in Table 4.2. rOSD generates proposals from local maxima of the image's saliency map obtained with CNN features. To evaluate rOSD+Self and rOSD+Sup for UOD, we assign each proposal a score equal to the saliency of the local maximum it is generated from. If two regions have the same score, the larger one is ranked higher so that entire objects instead of object parts are selected. Finally, when [Zitnick, 2014] proposals are used for rOSD and LOD, we multiply their features with their Edgeboxes scores before computing their similarity.

In general, variants with supervised features perform better in UOD than those with self-supervised features, except for [Wei, 2019] and LOD in single-object discovery on Op50K. [Kim, 2009] is the most dependent on supervised features. Its performance drops by at least 63% when switching to self-supervised features. It is also noteworthy that the performance of rOSD and LOD with supervised and self-supervised features on Op50K is much closer than on C20K. This is likely due to the fact that the supervised features [Simonyan, 2015a] are trained on the 1000 ImageNet object classes which contain all of the COCO classes and thus offer a stronger bias toward these classes than the self-supervised features. Using self-supervised features, variants of LOD are the best performer in both single-object discovery (with rOSD+Self proposals) and multi-object discovery (with Edgeboxes proposals). They yield reasonable results on both datasets compared to variants with supervised features. In particular, self-supervised object proposals (from rOSD) and self-supervised features, combined with LOD, give the best results of all tested methods on Op50K in single-object discovery. These results show that LOD combined with self-supervised features is a viable option for UOD without any supervision whatsoever.

**Comparing ranking formulations.** We compare the UOD performance of (Q), (P) and LOD with different proposals and features in Table 4.3. It can be seen that LOD outperforms (Q) and (P) in almost all datasets and settings. These results confirm the merit of our proposed

| Opt. | Proposal | Feature | Single-object | | Multi-object | | | |
|---|---|---|---|---|---|---|---|---|
| | | | CorLoc | | odAP50 | | odAP[50:95] | |
| | | | C20K | Op50K | C20K | Op50K | C20K | Op50K |
| None | [Zitnick, 2014] | None | 28.8 | 32.7 | 4.86 | 5.46 | 1.41 | 1.53 |
| | rOSD+Self | | 29.7 | 39.8 | 2.47 | 3.72 | 0.61 | 1.0 |
| | rOSD+Sup | | 23.6 | 38.1 | 4.07 | 4.81 | 1.03 | 1.39 |
| [Wei, 2019] | None | Self | 37.9 | 42.4 | 2.53 | 3.13 | 0.69 | 0.9 |
| | | Sup | 38.2 | 34.8 | 2.41 | 1.86 | 0.73 | 0.6 |
| [Kim, 2009] | [Zitnick, 2014] | Self | 5.5 | 5.4 | 0.64 | 0.79 | 0.13 | 0.15 |
| | | Sup | 15.6 | 20.2 | 1.96 | 2.56 | 0.36 | 0.47 |
| | rOSD+Self | Self | 4.7 | 4.6 | 0.13 | 0.29 | 0.02 | 0.05 |
| | rOSD+Sup | Sup | 35.1 | 37.0 | 3.93 | 4.13 | 0.96 | 0.98 |
| rOSD | [Zitnick, 2014] | Self | 35.6 | 43.6 | 3.34 | 4.43 | 0.99 | 1.39 |
| | | Sup | 40.2 | 44.0 | 4.0 | 4.47 | 1.21 | 1.41 |
| | rOSD+Self | Self | 37.8 | 48.1 | 2.65 | 4.19 | 0.82 | 1.45 |
| | rOSD+Sup | Sup | **48.5** | 48.0 | 5.18 | 4.98 | 1.62 | 1.58 |
| LOD | [Zitnick, 2014] | Self | 35.5 | 39.7 | 5.87 | 6.73 | 1.57 | 1.76 |
| | | Sup | 38.9 | 41.3 | 6.52 | **7.01** | 1.76 | 1.86 |
| | rOSD+Self | Self | 41.1 | **49.5** | 4.56 | 6.37 | 1.29 | 1.87 |
| | rOSD+Sup | Sup | **48.5** | 48.1 | **6.63** | 6.46 | **1.98** | **1.88** |

Table 4.2 – UOD performance with supervised [Simonyan, 2015a] (Sup) and self-supervised [Gidaris, 2021] (Self) features on C20K and Op50K datasets. Region proposals are generated by methods from Edgeboxes[Zitnick, 2014] and rOSD with different types of features. LOD with self-supervised features yields reasonable results compared to supervised features. Variants of our proposed method LOD yield state-of-the-art performance in all settings.

method, using (Q)'s solution to personalize PageRank.

**Influence of hyper-parameters.** The proposed method has two important hyper-parameters, the damping factor $\beta$ in PageRank and the scalar $\alpha$ used to select reliable object candidates in LOD. In practice, $\beta$ should be small so as not to change much the weight matrix $A$ and $\alpha$ should also be small since we only want to select a few top-scoring proposals. We have evaluated PageRank for object discovery on C20K and Op50K datasets with increasing values of $\beta$, ranging from $10^{-5}$ to $10^{-1}$, and present the results in Table 4.4 (left). This experiment shows that the performance of PageRank begins to drop when $\beta$ becomes larger than $10^{-3}$ and deteriorates significantly when it exceeds $10^{-2}$. It does not depend much on $\beta$ when this parameter is small enough (less than $10^{-3}$). We choose $\beta = 10^{-4}$ in our implementation. We have also evaluated LOD with different values of $\alpha$, taken in $\{0.05, 0.1, 0.15, 0.2\}$, which amounts to selecting 5%, 10%, 15% and 20% of candidates respectively, and show the results in Table 4.4 (right). As long as $\alpha$ is reasonably small, its value does not significantly affect the performance of LOD. We choose $\alpha = 0.1$ in our implementation.

**Varying the number of region proposals.** Unlike rOSD, we are able to use in LOD almost all the regions produced by the proposal algorithm (2000 regions per image at most) thanks to

| Opt. | Proposal | Feature | Single-object | | Multi-object | | | |
|---|---|---|---|---|---|---|---|---|
| | | | CorLoc | | odAP50 | | odAP[50:95] | |
| | | | C20K | Op50K | C20K | Op50K | C20K | Op50K |
| (Q) | [Zitnick, 2014] | Self | 32.8 | 40.3 | 4.15 | 6.43 | 1.07 | 1.67 |
| | | Sup | 36.0 | 41.1 | 5.72 | 6.49 | 1.47 | 1.7 |
| | rOSD+Self | Self | 38.7 | 48.9 | 4.38 | 6.39 | 1.17 | 1.84 |
| | rOSD+Sup | Sup | 43.8 | 47.5 | 6.21 | 6.66 | 1.74 | **1.88** |
| (P) | [Zitnick, 2014] | Self | 35.5 | 39.7 | 4.91 | 6.73 | 1.34 | 1.75 |
| | | Sup | 38.9 | 41.3 | 6.51 | 6.99 | 1.76 | 1.86 |
| | [Vo, 2020]+Self | Self | 41.2 | **49.5** | 4.38 | 6.13 | 1.24 | 1.81 |
| | [Vo, 2020]+Sup | Sup | 47.5 | 47.8 | 6.25 | 6.19 | 1.87 | 1.81 |
| LOD | [Zitnick, 2014] | Self | 35.5 | 39.7 | 5.87 | 6.73 | 1.57 | 1.76 |
| | | Sup | 38.9 | 41.3 | 6.52 | **7.01** | 1.76 | 1.86 |
| | rOSD+Self | Self | 41.1 | **49.5** | 4.56 | 6.37 | 1.29 | 1.87 |
| | rOSD+Sup | Sup | **48.5** | 48.1 | **6.63** | 6.46 | **1.98** | **1.88** |

Table 4.3 – A comparison of different ranking methods for UOD. LOD is better than (Q) and (P) in most of the cases.

| $\beta$ | Single-object | | Multi-object | | | |
|---|---|---|---|---|---|---|
| | CorLoc | | odAP50 | | odAP[50:95] | |
| | C20K | Op50K | C20K | Op50K | C20K | Op50K |
| $10^{-5}$ | 48.0 | 47.8 | 6.3 | 6.13 | 1.89 | 1.8 |
| $10^{-4}$ | 48.0 | 47.8 | 6.29 | 6.19 | 1.89 | 1.81 |
| $10^{-3}$ | 47.9 | 47.7 | 6.22 | 6.08 | 1.87 | 1.78 |
| $10^{-2}$ | 47.0 | 47.0 | 5.82 | 5.69 | 1.76 | 1.68 |
| $10^{-1}$ | 40.0 | 38.8 | 4.45 | 4.14 | 1.34 | 1.22 |

| $\alpha$ | Single-object | | Multi-object | | | |
|---|---|---|---|---|---|---|
| | CorLoc | | odAP50 | | odAP[50:95] | |
| | C20K | Op50K | C20K | Op50K | C20K | Op50K |
| 0.05 | 48.4 | 48.2 | 6.63 | 6.5 | 1.99 | 1.89 |
| 0.10 | 48.5 | 48.1 | 6.63 | 6.46 | 1.98 | 1.88 |
| 0.15 | 48.5 | 48.2 | 6.64 | 6.49 | 1.99 | 1.89 |
| 0.20 | 48.5 | 48.2 | 6.64 | 6.48 | 1.99 | 1.89 |

Table 4.4 – Influence of the damping factor $\beta$ on PageRank's performance (left) and of the selection factor $\alpha$ on LOD's performance (right) on the C20K and Op50K datasets.

the good scalability of our formulation. On average, we have 814 and 850 regions per image on C20K and Op50K, respectively. We have evaluated LOD on C20K and Op50K using different numbers of proposals and observed in Table 4.5 that its performance improves with additional region proposals, notably in the multi-object setting. When a subset of only 100 regions is used, the odAP50 of LOD matches the top performance of rOSD, and increases with more regions. This observation partly explains our better performance compared to rOSD (which places a limit on the number of regions for computational reasons) and the benefit of using all region proposals.

**Influence of underlying features.** We use features from a VGG16 [Simonyan, 2015a] model trained for image classification on ImageNet [Deng, 2009] in our main experiments. We have also tested LOD with features from VGG19 [Simonyan, 2015a] and ResNet50 [He, 2016] and present the results on C20K and Op50K in Table 4.6. Although VGG19 and ResNet50 give better results in image classification [Simonyan, 2015a; He, 2016], they perform worse than VGG16 in object discovery with LOD. This may be due to the fact that they are more discriminative,

| Num. of regions | C20K | | | Op50K | | |
|---|---|---|---|---|---|---|
| | CorLoc | odAP50 | odAP[50:95] | CorLoc | odAP50 | odAP[50:95] |
| 50 | 40.9 | 4.5 | 1.22 | 42.0 | 4.55 | 1.31 |
| 100 | 44.0 | 5.38 | 1.47 | 43.4 | 5.1 | 1.4 |
| 200 | 46.5 | 6.13 | 1.71 | 45.6 | 5.83 | 1.61 |
| 400 | 48.0 | 6.6 | 1.91 | 47.1 | 6.32 | 1.77 |
| All | 48.5 | 6.63 | 1.98 | 48.1 | 6.46 | 1.88 |

Table 4.5 – The performance of LOD on C20K and Op50K datasets when varying the number of region proposals per image. Using more regions improves LOD's performance.

focusing mostly on the most prominent object parts thus less helpful in localizing entire objects, although we do not have a definitive answer (yet) for this.

| Features | Single-object | | Multi-object | | | |
|---|---|---|---|---|---|---|
| | CorLoc | | odAP50 | | odAP[50:95] | |
| | C20K | Op50K | C20K | Op50K | C20K | Op50K |
| VGG19 | 47.4 | 45.1 | 6.27 | 5.57 | 1.84 | 1.58 |
| ResNet50 | 35.4 | 45.9 | 4.08 | 5.59 | 1.05 | 1.46 |
| VGG16 | **48.5** | **48.1** | **6.63** | **6.46** | **1.98** | **1.88** |

Table 4.6 – LOD performance with VGG [Simonyan, 2015a] and ResNet50 [He, 2016] features on C20K and Op50K datasets. Although the latter are more powerful in image classification, VGG16 features yield the best results in object discovery with LOD.

### 4.4.2 Category Discovery

Contrary to [Cho, 2015], OSD and rOSD, our work aims specifically at localizing objects in images and omits the discovery of the image graph structure, i.e., identifying image pairs that contain objects of the same category. However, objects localized by our methods can be used to perform this task in a post-processing step. To this end, we define similarity between two images as the maximum similarity between pairs of selected proposals. Similarity is measured using cosine distance between features extracted from the *fc6* layer. We compare LOD to [Cho, 2015], OSD and rOSD in image neighbor retrieval task on VOC_all, the subset of Pascal VOC2007 dataset [Everingham, 2007] used as a benchmark in [Cho, 2015] and the previous chapters. Similar to these works, we retrieve 10 nearest neighbors per image. Then, CorRet ('any' version) [Cho, 2015] – the average percentage of retrieved image neighbors that are actual neighbors in the ground-truth image graph over all images – is used to compare different methods. Results are shown in Figure 4.5. LOD outperforms [Cho, 2015], OSD and rOSD. This is surprising since the other methods are specifically formulated to discover image neighbors, while our method is not. This result highlights that our localized objects can be potentially beneficial for other tasks.

To go further, we cluster images into categories using proposals selected by our algorithm. Imposing that images are represented by their proposal with the highest score, we perform this

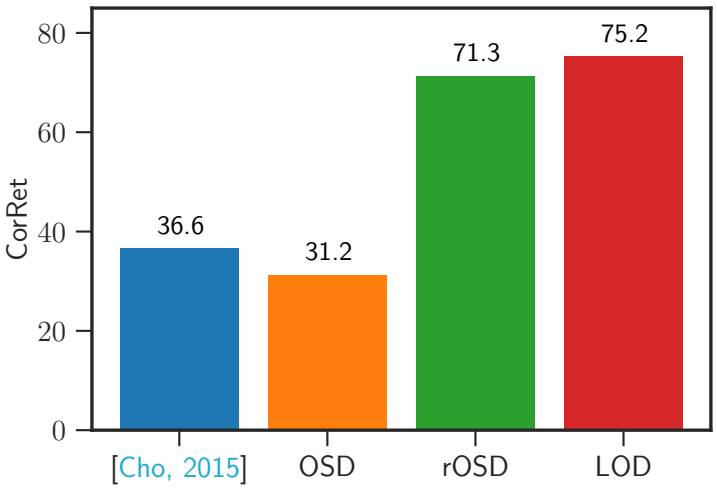



Figure 4.5 – Comparison to prior work on the image neighbor retrieval task using CorRet measure. Higher is better.

| Dataset | LOD | [Zhang, 2015b] | [Zhu, 2012] | [Feng, 2011] | [Zhang, 2011] | [Zhang, 2009] | [Kim, 2008] | [Kim, 2012] |
|---|---|---|---|---|---|---|---|---|
| SIVAL1 | **97.4** | 89.0 | 95.3 | 80.4 | 39.3 | 38.0 | 27.0 | 45.0 |
| SIVAL2 | **99.0** | 93.2 | 84.0 | 71.7 | 40.0 | 33.3 | 35.3 | 33.3 |
| SIVAL3 | 88.3 | **88.4** | 74.7 | 62.7 | 37.3 | 38.7 | 26.7 | 41.3 |
| SIVAL4 | **97.7** | 87.8 | 94.0 | 86.0 | 33.0 | 37.7 | 27.3 | 53.0 |
| SIVAL5 | **94.3** | 92.7 | 75.3 | 70.3 | 35.3 | 37.7 | 25.0 | 48.3 |
| Average | **95.3** | 90.2 | 84.7 | 74.2 | 37.0 | 37.1 | 28.3 | 44.2 |

Table 4.7 – Purity (↑) of our clustering method compared to the state of the art in category discovery on the SIVAL dataset [Rahmani, 2008]. Following prior work, we perform the task on a partition of the dataset and report the average purity on its parts as the final result. Results of other methods are from [Zhang, 2015b].

task by applying $K$-means on the $L_2$-normalized *fc6* features representing these proposals. We conduct experiments on the SIVAL [Rahmani, 2008] dataset, a popular benchmark for this task. This dataset consists of 25 object categories, each containing about 60 images. Following [Zhu, 2012], we partition the 25 object classes into 5 groups, named SIVAL1 to SIVAL5, and use purity (average percentage of the dominant class in the clusters) as an evaluation metric. Intuitively, purity measures the extent to which a cluster contains images of a single dominant class. A comparison between our method and other popular object category discovery methods is given in Table 4.7. It can be seen that our method outperforms the state of the art by a significant margin, attaining an average purity of 95.3. It is also noteworthy that the performance drops to 23.7 when the features of entire images are used instead of the representative top proposals. This finding shows that our performance gain is in great part due to the object localization performance of our method.

Since individual images in the SIVAL dataset [Rahmani, 2008] contain only one object, we conduct a similar experiment on the more challenging VOC_all [Everingham, 2007] dataset. In this experiment, a histogram is computed for each cluster, showing the score of each ground-truth object category (a category score is the sum of contributions of all its images). An image

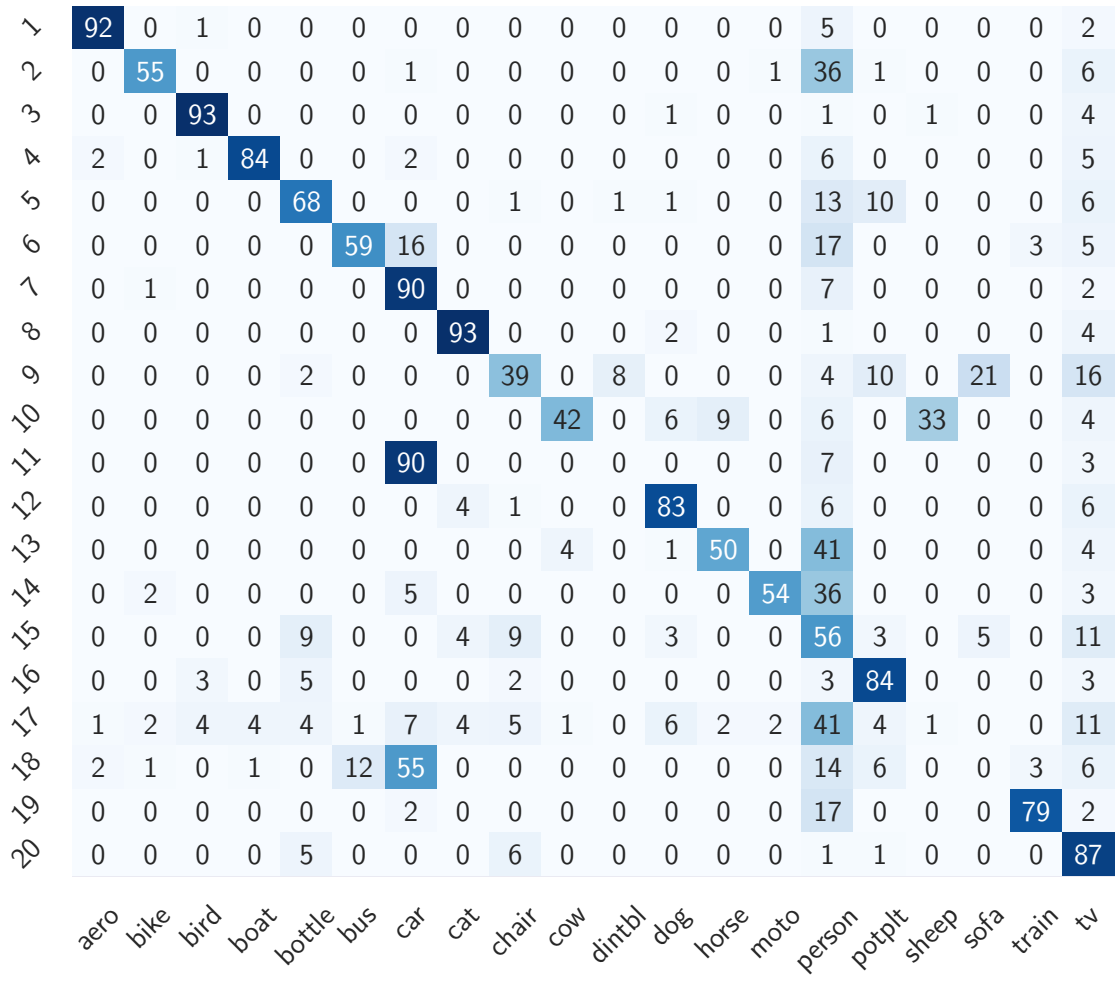



Figure 4.6 – Confusion matrix revealing links between the object classes and the clusters found by LOD on VOC_all.

contribution is computed as $1/nc$, where $c$ is the number of object categories appearing in the image and $n$ is the number of images in the cluster. We then match the clusters to the ground-truth categories by solving a stable marriage problem with the Gale–Shapley algorithm [Gale, 1962] using the preference orders induced by the histograms. The confusion matrix generated by combining these histograms, revealing the correspondence between the clusters and the classes, is shown in Figure 4.6. Our method is able to discover 17 categories (which are dominant in at least one cluster) out of 20 ground-truth categories. As for the three undiscovered categories: *sheep* is dominated by similar class *cow* in cluster 10; *sofa* is dominated by co-occurring class *chair* in cluster 9; *dinningtable* suffers from being often largely occluded in images. Interestingly, it seems that our method might be used to discover pairs of categories that often appear together, for instance: *bicycle* and *person*, *horse* and *person*, *motorbike* and *person* (clusters 2, 13 and 14 have two corresponding dominating classes each). Quantitatively, using the top extracted proposals from our method achieves a purity of 68.6 on this dataset, which is better than the purity of 61.8 obtained when features of entire images are used.

### 4.4.3 Discussions

Without a formal definition of objects, casting objects as frequently appearing salient visual patterns is natural. However, findings could be biased toward popular object classes and ignore rare classes in image collections that contain a long-tail distribution of object classes. To have an insight to this potential bias, we compute LOD's performance by object category on C20K dataset. Surprisingly, we have observed little correlation between the performance on an object class and its appearance frequency (the corresponding correlation is only $-0.09$, see Figure 4.7). A possible explanation is that even though we rank all regions in the image collection at once, we choose objects (based on the ranking) on the image level. Therefore, regions can be selected as objects if they stand out more from the background and are better connected in the graph than other regions in the same image, even if they represent objects of a rare class.

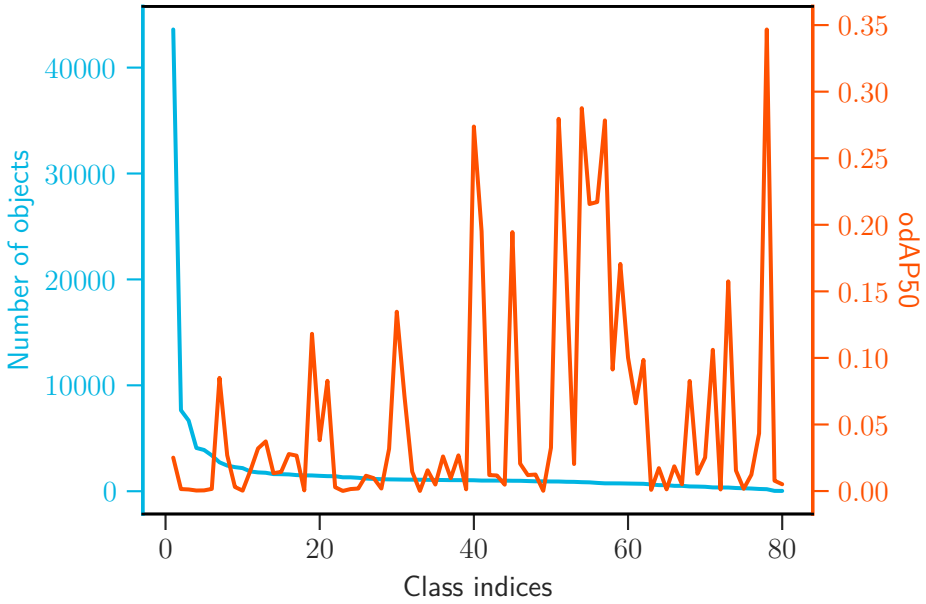



Figure 4.7 – Performance of LOD by object category and category frequency (number of object occurrences of each category) on the C20K dataset. Results are reported with odAP50, higher numbers are better. Object categories are indexed in decreasing order of category frequency. Performance of LOD is not well-correlated (correlation $-0.09$) with category frequency.

## 4.5 Conclusion and Future Work

We have demonstrated a novel formulation of unsupervised object discovery as a ranking problem, allowing application of efficient and distributed algorithms used for link analysis and ranking problems. In particular, we have shown how to apply the personalized PageRank algorithm to derive a solution to UOD, and proposed a new technique based on eigenvector computation to identify the personalized vector in Pagerank. The proposed LOD algorithm naturally admits a distributed implementation and allows us to scale up UOD to the OpenImages [Krasin, 2017] dataset (Op1.7M) with 1.7M images, 87 times larger than datasets considered in rOSD,

and outperforms (in single- and multi-object discovery) all existing algorithms capable of scaling to this size. In multi-object discovery, LOD is better than all other methods on medium and large-scale datasets. State-of-the-art solutions to UOD rely on supervised region proposals [Cho, 2015] or features ([Wei, 2019] and rOSD), thus their output requires at least in part some sort of supervision. We have proposed to combine LOD with self-supervised features, offering a solution to fully unsupervised object discovery. Finally, we have shown that LOD yields state-of-the-art results in category discovery, which is obtained as a post-processing step.

Similar to OSD and rOSD, in LOD, we define objects as salient visual patterns that appear in multiple images and find them by considering the pairwise similarity between region proposals in different images. This results in a high computational cost due to the large number of region pairs to take into account. In the next chapter, we show that we can tackle unsupervised object discovery without considering pairwise region similarity, thereby avoid its entailed cost, by exploiting the recent transformer-based self-supervised features DINO [Caron, 2021].

# Chapter 5

# Localizing Objects with Self-Supervised Transformers and no Labels

Objectives

We propose in this chapter a simple method that leverages the activation features of a vision transformer pre-trained in a self-supervised manner to localize objects in images. Our algorithm, LOST, does not require any external object proposal nor any exploration of the image collection; it operates on a single image. Yet, LOST outperforms state-of-the-art object discovery methods by up to 8 CorLoc points on PASCAL VOC 2012. We also show that training a class-agnostic detector on the discovered objects boosts results by another 7 points. Moreover, we show promising results on the unsupervised object detection task.
This work, done in collaboration with Oriane Siméoni, Gilles Puy, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Renaud Marlet, Patrick Pérez and Jean Ponce, has appeared in Proceedings of the British Machine Vision Conference (BMVC) 2021.

## Contents


5.1 Introduction . . . . . **76**
5.2 Related Work . . . . . **77**
5.3 Proposed Approach . . . . . **78**
5.3.1 Transformers for Vision . . . . . 78
5.3.2 Finding Objects with LOST . . . . . 79
5.3.3 Towards Unsupervised Object Detection . . . . . 81
5.4 Experiments . . . . . **82**
5.4.1 Experimental Setup . . . . . 82
5.4.2 Single-Object Discovery . . . . . 83
5.4.3 Unsupervised Object Detection . . . . . 84
5.4.4 Ablation Study . . . . . 87
5.5 Conclusion, Limitations and Future Work . . . . . **90**

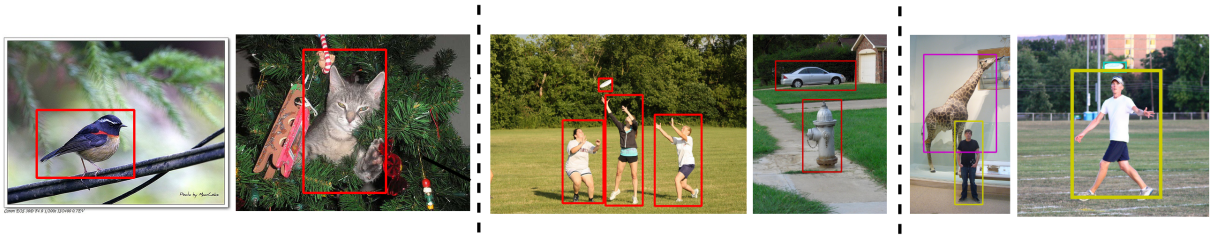


Figure 5.1 – Three applications of LOST to unsupervised single-object discovery (left), multi-object discovery (middle) and object detection (right). In the latter case, objects discovered by LOST are grouped into clusters, and cluster indices are used to train a classical object detector. Although large image collections are used as pseudo labels to train the underlying image representation [Caron, 2021], *no annotation* is ever used in the pipeline. See Tables 5.1 and 5.3, and Figure 5.3 for more experiments.

## 5.1 Introduction

We propose in this chapter LOST, a simple approach to localizing objects in an image, that we then apply to unsupervised object discovery. Our localization method stays at the level of a single image, rather than exploring inter-image similarity, which makes it linear w.r.t. the number of images, avoiding the expensive computational cost of OSD, rOSD and LOD. For this, we leverage high-quality features obtained from DINO, a visual transformer pre-trained with self-supervision [Caron, 2021]. Concretely, we divide the image of interest into equal-sized patches and feed it to the DINO model. Instead of focusing on the `CLS` token, we propose to use the *key* component of the last attention layer for computing the similarities between the different patches. In doing so, we are able to localize a part of an object by selecting the patch with the least number of similar patches, here called the *seed*. The justification for this seed selection criterion is based on the empirical observation that patches of foreground objects are less correlated than patches corresponding to background. We add to this initial seed other patches that are highly correlated to it and thus likely to be part of the same object, a process akin to query expansion in the retrieval literature, which we call *seed expansion*. Finally, we construct a binary object segmentation mask by computing the similarities of each patch to the selected seed patches and infer a tightly box around the mask's largest connected component that contains the initial seed. In following this simple method, we not only outperform methods for region proposals but also those for single-object discovery.

We go further by training an off-the-self class-agnostic object detector using our localized boxes as pseudo ground-truth boxes, and are able to derive a much more accurate object localization model that is able to discover multiple objects in an image. We call this task *unsupervised class-agnostic object detection*. Finally, by using clustering techniques to group the localized objects into visually consistent classes, we are able to train class-aware object detectors without any human supervision, but using instead the predicted object locations and their cluster indices as ground-truth annotations. We call this task *unsupervised (class-aware) object detection*. We show that the predictions of our unsupervised detection model for certain clusters correlate

very well with labelled semantic classes in the dataset and achieve with them detection results competitive weakly-supervised object detectors [Bilen, 2016; Tang, 2018c]. Note that our unsupervised class-aware object detector gives the standard output for object detection (rectangular boxes and their class scores) but, with a score threshold, we can easily transform this output to the standard format for unsupervised object discovery, i.e., predicted object boxes and image pairs that contain similar objects. Our main contributions are as follows:

- We show how to extract relevant features from a self-supervised pre-trained vision transformer and use the patch correlations within an image to propose a simple single-object localization method with linear complexity with respect to the dataset size.
- We leverage it to train both class-agnostic and class-aware unsupervised object detectors that are able to accurately localize multiple objects per image and, in the class-aware case, group them into semantically-coherent classes.
- We outperform the state of the art in unsupervised object discovery by a significant margin.

## 5.2 Related Work

**Transformers.** In this chapter, we leverage transformer representations to address object discovery. Self-attention layers have been previously integrated into CNNs [Hu, 2018; Wang, 2018c; Carion, 2020], yet transformers for vision are recent [Ramachandran, 2019; Cordonnier, 2020; Chen, 2020a; Dosovitskiy, 2021] and still in an incipient stage. Findings on training heuristics [Touvron, 2020; Zhai, 2021] and architecture design [Liu, 2021b; Touvron, 2021; Yuan, 2021a] are released at high pace. Early adaptations of transformers to different tasks (*e.g.*, image classification [Dosovitskiy, 2021], retrieval [ElNouby, 2021], object detection [Carion, 2020; Zhu, 2021; Liu, 2021b] and semantic segmentation [Liu, 2021b; Strudel, 2021; Xie, 2021a] have demonstrated their utility and potential for vision. Meanwhile, several works attempt to better understand this new family of models from various perspectives [Caron, 2021; Naseer, 2021; Tuli, 2021; Bhojanapalli, 2021; Minderer, 2021]. Interestingly, transformers have been shown to be less biased towards textures than CNNs [Tuli, 2021; Naseer, 2021], hinting that their features encapsulate more object-aware representations. These findings motivate us to study ways of localizing objects from transformer features.

**Self-supervised learning (SSL)** is a powerful training scheme to learn useful representations without human annotations. It does so via a pretext learning task for which the supervision signal comes from the data itself [Noroozi, 2016; Gidaris, 2018b; Zhang, 2016]. SSL pre-trained networks have been shown to outperform ImageNet pre-trained networks on several computer vision tasks, in particular object detection [Gidaris, 2020; He, 2020; Caron, 2020; Grill, 2020; Gidaris, 2021]. For transformers, SSL methods also work well [Caron, 2021; Xie, 2021b], bringing a few interesting side-effects. In particular, DINO [Caron, 2021] feature activations appear to contain explicit information about the location of objects in an image. In the same spirit, we extract another kind of transformer features to build our object localization model.

**Object detection with limited supervision.** Region proposal methods [Alexe, 2012; Uijlings, 2013; Zitnick, 2014] generate in an unsupervised way numerous class-agnostic bounding boxes with high recall but low precision, to speed-up sliding window search. From supervised pre-trained networks, objects can emerge by masking the input [Bergamo, 2016], interpreting neurons [Zhou, 2015] or from saliency maps [Selvaraju, 2017]. Weakly-supervised object detection (WSOD) uses image-level labels without bounding boxes [Bilen, 2016; Tang, 2018c] to learn to detect objects. The different instances of WSOD (each with specific assumptions on the availability and amount of image-level and box-level annotations) are often addressed as semi-supervised learning [Gao, 2019a; Tang, 2021a] and leverage self-training [Radosavovic, 2017; Jie, 2017b]. Recent work replaces manual annotations with automatic supervision from a different modality, *e.g.*, LiDAR [Tian, 2021] or audio [Afouras, 2021]. In contrast, we do not use any annotations or other modalities at any stage: we extract object candidates from the activations of a self-supervised pre-trained network, compute pseudo-labels and then train an object detector.

## 5.3 Proposed Approach

Our method exploits image representations extracted from a vision transformer. In this section, we first recall how such representations are obtained, then present our method.

### 5.3.1 Transformers for Vision

**Input.** Vision transformers operate on a sequence of patches of fixed size $P \times P$. For a color image $\mathbf{I}$ of spatial size $H \times W$, we have $N = HW/P^2$ patches of size $P \times P \times 3$ (we assume for simplicity that $H$ and $W$ are multiples of $P$). Each patch is first embedded in a $d$-dimensional latent space via a trained linear projection layer. An additional, learned vector called the "class token", `CLS`, is adjoined to the patch embeddings, yielding a transformer input in $\mathbb{R}^{(N+1)\times d}$.

**Self-attention.** Transformers consist of a sequence of multi-head self-attention layers and multi-layer perceptrons (MLPs) [Vaswani, 2017; Dosovitskiy, 2021]. Three different learned linear transformations are applied to an input $\mathbf{X} \in \mathbb{R}^{(N+1)\times d}$ of a self-attention layer to produce a *query* $\mathbf{Q}$, a *key* $\mathbf{K}$ and a *value* $\mathbf{V}$, all in $\mathbb{R}^{(N+1)\times d}$. The output of the self-attention layer is $\mathbf{Y} = \text{softmax}\left(d^{-1/2}\,\mathbf{Q}\mathbf{K}^\top\right)\mathbf{V} \in \mathbb{R}^{(N+1)\times d}$, where softmax is applied row-wise. For simplicity, we describe here the case of a single-head attention layer, but attention layers usually contain multiple heads. In this work, we concatenate the keys (or queries, or values) from all heads in the last self-attention layer to obtain our feature representations.

**Features for object localization.** We use transformers trained in a self-supervised manner with DINO [Caron, 2021]. In this work, the authors show that reasonable object segments can be obtained from the self-attention of the `CLS` query produced by the last attention layer. We adapt this strategy in Section 5.4 to perform object localization, providing a baseline ('DINO-seg') that produces fair results. However, we found that its does not fully exploit the potential of the transformer features. We propose a novel and effective strategy for localizing objects

using another way to extract and use features. Our method, called LOST, is constructed by computing similarities between patches of a single image, using this time patch keys $\mathbf{k}_p \in \mathbb{R}^d$, $p = 1, \dots, N$, extracted at the last layer of a DINO transformer.

### 5.3.2 Finding Objects with LOST



Figure 5.2 – Initial seed, patch similarities and patch degrees. Each column corresponds to one image. The top row shows original images from the Pascal VOC2007 dataset. The images in the middle row illustrate the initial seed $p^*$ (in red) and patches similar to $p^*$ (in grey), *i.e.*, patches $q$ such that $\mathbf{f}_{p^*}^\top \mathbf{f}_q \geq 0$. The bottom row shows maps of the inverse degree $1/d_p$ of all patches $p$ (yellow corresponds to high inverse degree and blue corresponds to low inverse degree). The initial seed $p^*$ is the patch with the lowest degree. Best viewed in color.

Our method takes as input $d$-dimensional image features $\mathbf{F} \in \mathbb{R}^{N \times d}$ extracted from a single image via a neural network, where $N$ denotes the the number of patches in the image, and $\mathbf{f}_p \in \mathbb{R}^d$ is the feature vector of the patch at spatial position $p \in \{1, \dots, N\}$. We assume that there is at least one object in the image and LOST tries to localize one such object given the input features. To this end, it relies on a selection of patches that are likely to belong to an object. We call these patches "seeds".

**Initial seed selection.** Our seed selection strategy is based on the assumptions that (a) regions/patches within objects correlate more with each other than with background patches and vice versa, and (b) an individual object covers less area than the background. Consequently, a patch with little correlation with the rest of the image has a higher chance to belong to an object.

To compute the patch correlations, we rely on the distinctiveness of self-supervised transformer features, which is particularly noticeable when using transformer keys. We empirically observe that using these tranformer features as patch representation meets assumption (a) in practice: patches in an object correlate positively with each other but negatively with patches

in the background. Therefore, based on assumption (b), we pick the patch with the smallest number of positive correlations with other patches as the initial seed $p^*$.

Concretely, we build a patch similarity graph $\mathcal{G}$ for each image, represented by the binary symmetric adjacency matrix $\mathbf{A} = (a_{pq})_{1 \leq p,q \leq N} \in \{0,1\}^{N \times N}$ such that

$$a_{pq} = \begin{cases} 1 & \text{if } \mathbf{f}_p^\top \mathbf{f}_q \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

In other words, two nodes $p, q$ are connected by an undirected edge if their features $\mathbf{f}_p, \mathbf{f}_q$ are not negativelyly correlated. We select the initial seed $p^*$ as a patch with the lowest degree $d_p$, that is:

$$p^* = \underset{p \in \{1,\dots,N\}}{\arg\min}\ d_p\ , \quad \text{where} \quad d_p = \sum_{q=1}^{N} a_{pq}. \tag{5.2}$$

We show in Figure 5.2 examples of seeds $p^*$ selected in four different images along with the degree maps. We remark that the patches with lowest degrees are the most likely to fall in an object. Finally, we also observe in this figure that the few patches that correlate positively with $p^*$ are also likely to belong to an object.

**Seed expansion.** Once the initial seed has been picked, we select other patches that are correlated with it and likely to belong to an object. Again, we rely on the empirical observations that pixels within an object tend to be positively correlated and have a small degree in $\mathcal{G}$. The selection is done by finding the set $\mathcal{S}$ of patches in $D_k$ – the set of $k$ patches with the lowest degree – whose features correlate positively with $\mathbf{f}_{p^*}$:

$$\mathcal{S} = \{q \mid q \in \mathcal{D}_k \text{ and } \mathbf{f}_q^\top \mathbf{f}_{p^*} \geq 0\}. \tag{5.3}$$

In case of patches with equal degrees, we break ties arbitrarily to ensure that $|\mathcal{D}_k| = k$ and we typically use $k = 100$ in our implementation.

**Box extraction.** The last step consists in computing a mask $\mathbf{m} \in \{0,1\}^N$ by comparing the features of seeds in $\mathcal{S}$ with the features of all other patches. The $q^{\text{th}}$ entry of the mask $\mathbf{m}$ satisfies

$$m_q = \begin{cases} 1 & \text{if } \sum_{s \in \mathcal{S}} \mathbf{f}_q^\top \mathbf{f}_s \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

In other words, a patch $q$ is considered as part of an object if, on average, the corresponding feature $\mathbf{f}_q$ positively correlates with the features of the patches in $\mathcal{S}$. To remove spurious correlated patches, we finally select the connected component in $\mathbf{m}$ that contains the initial seed and use the bounding box of this component as the discovered object. An illustration of the discovered objects before and after seed expansion is provided in Figure 5.3.

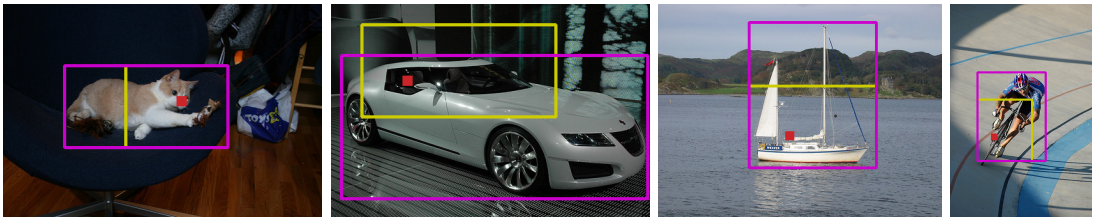


Figure 5.3 – Objects discovered by LOST on VOC07. The red square represents the seed $p^*$, the yellow box is the box obtained using only $p^*$, and the purple box is the one obtained using all the seeds in $\mathcal{S}$, collected via seed expandion. Using only the initial seed $p^*$, the returned boxes tend to focus only on the most discriminative parts of objects. Seed expansion allows returning larger regions that cover the entire object extent.

### 5.3.3 Towards Unsupervised Object Detection

We exploit the accurate single-object localization of LOST for training object detectors without any human supervision. Starting from a set of unlabeled images, each one assumed to contain at least one prominent object, we extract one bounding box per image using LOST. Then, we train object detectors using these pseudo-annotated boxes. We explore two scenarios: class-agnostic and (pseudo) class-aware training of object detectors.

**Class-agnostic detection (CAD).** A class-agnostic detection model localizes salient objects in an image without predicting nor caring about their semantic category. We train such a detector by assigning the same "foreground" category to all the boxes produced by LOST, which we call "pseudo-boxes" afterwards, as they are obtained automatically without human effort. Unlike LOST, the trained detector can localize multiple objects per image, even if it was trained on a dataset containing only one pseudo-box annotation per image. The experiments confirm that the trained detector can output multiple detections and the quantitative results (Table 5.1) show that this trained detector is in fact even better than LOST in terms of localization accuracy.

**Class-aware detection (OD).** We now consider a typical detector that both localizes objects and recognizes their semantic category. To train such a detector, apart from LOST's pseudo-boxes, we also need a class label for each of these boxes. In order to remain fully-unsupervised, we discover visually-consistent object categories using K-means clustering. For each image, we crop the object detected by LOST, resize the cropped image to $224 \times 224$, feed this image in the DINO pre-trained transformer, and extract the `CLS` token at the last layer. The set of `CLS` tokens are clustered using K-means and the cluster index is used as a pseudo-label for training the detector. At evaluation time, we match these pseudo-labels to the ground-truth class labels using the Hungarian algorithm [Kuhn, 1955], which give names to pseudo-labels.

## 5.4 Experiments

We explore in this section three variants of the object localization problem, in order of increasing complexity: (1) localizing one salient object in each image (single-object discovery) in

Section 5.4.2, (2) using the corresponding bounding boxes as ground-truth to train a binary classifier for foreground object detection (unsupervised class-agnostic object detection), and (3) using clustering to capture an unsupervised notion of object categories, and detect the corresponding instances (unsupervised object detection). The last two variants are discussed in Section 5.4.3. None of the building blocks of this pipeline uses any annotation, just a large number of unlabelled images to sequentially train, in a self-supervised way, the DINO transformer, the class-agnostic foreground/background classifier, and finally the classifier using the cluster identifiers as labels.

### 5.4.1 Experimental Setup

**Backbone networks.** Unless otherwise specified, we use the self-supervised ViT-S model introduced in [Caron, 2021], which follows the architecture of DeiT-S [Touvron, 2020]. It is trained using DINO [Caron, 2021], with a patch size of $P = 16$ and the keys $\mathbf{K}$ (without the entry corresponding to the `CLS` token) of the last layer as input features $\mathbf{F}$, with which we achieve the best results. Results obtained alternatively with the attention, queries and values are presented and discussed Section 5.4.4. For comparison, we also present results using the base version of ViT (ViT-B), ViT-S with a patch size of $P = 8$, as well as with features of the last convolutional layer of a dilated ResNet-50 [He, 2016] and of a VGG16 [Simonyan, 2015a] pre-trained either following DINO, or in a supervised fashion on Imagenet [Deng, 2009].

**Datasets.** We evaluate the performance of our approach on the three variants of object localization on VOC07 [Everingham, 2007] trainval+test, VOC12 [Everingham, 2012] trainval and C20K. VOC07 and VOC12 are commonly used benchmarks for object detection [Girshick, 2014; Girshick, 2015]. C20K is a subset of the COCO2014 trainval dataset [Lin, 2014], consisting of 19817 randomly chosen images, used as a benchmark in rOSD (see Chapter 3). When evaluating results on the unsupervised object discovery task, we follow a common practice and evaluate scores on the trainval set of the different datasets. Such an evaluation is possible as the task is fully unsupervised. We follow the same principle for the unsupervised class-agnostic task: we generate boxes on VOC07 trainval, VOC12 trainval and C20K, use them to train a class-agnostic detector, and then evaluate again on these datasets (against ground-truth boxes this time).

For unsupervised class-aware object detection, we generate boxes and train the detector on VOC07 trainval and/or VOC12 trainval, but evaluate the detector on the VOC07 test set to facilitate comparisons to weakly-supervised object detection methods. Note that for unsupervised object discovery, in the previous chapters, we have used instead subsets of VOC07 trainval and VOC12 trainval for evaluation. For completeness, we also present the object discovery performance of LOST on these smaller subsets in Appendix D.

### 5.4.2 Single-Object Discovery

Similar to methods for unsupervised single-object discovery, LOST produces one box for each image. It therefore can be directly evaluated for this task.

| Method | VOC07_trainval | VOC12_trainval | C20K |
|---|---|---|---|
| [Uijlings, 2013] | 18.8 | 20.9 | 16.0 |
| [Zitnick, 2014] | 31.1 | 31.6 | 28.8 |
| [Kim, 2009] | 43.9 | 46.4 | 35.1 |
| [Zhang, 2020d] | 46.2 | 50.5 | 34.8 |
| [Wei, 2019] | 50.2 | 53.1 | 38.2 |
| rOSD | 54.5 | 55.3 | 48.5 |
| LOD | 53.6 | 55.1 | 48.5 |
| DINO-seg (w. ViT-S/16) | 45.8 | 46.2 | 42.1 |
| LOST (ours) | **61.9** | **64.0** | **50.7** |
| rOSD + CAD | 58.3 | 62.3 | 53.0 |
| LOD + CAD | 56.3 | 61.6 | 52.7 |
| LOST + CAD | **65.7** | **70.4** | **57.5** |

Table 5.1 – Single-object discovery performance in CorLoc on VOC07 trainval, VOC12 trainval and C20K. We compare LOST to state-of-the-art object discovery methods [Kim, 2009; Wei, 2019; Zhang, 2020, rOSD, LOD] as well as to two object proposal methods [Uijlings, 2013; Zitnick, 2014]. We also compare to the segmentation method proposed in DINO [Caron, 2021], denoted by DINO-seg. Additionally, we train a class-agnostic dectector (+ CAD) using as ground-truth either our pseudo-boxes or the boxes of rOSD or LOD.

**Comparison to prior work.** In Table 5.1, we present the CorLoc of our method, in comparison to state-of-the-art object discovery methods [Kim, 2009; Wei, 2019; Zhang, 2020, rOSD, LOD] and region proposals [Uijlings, 2013; Zitnick, 2014]. Despite its simplicity, we see that LOST outperforms the other methods by large margins. We also compare against an adapted version of the segmentation method proposed in [Caron, 2021]. Concretely, we extract the self-attention of the `CLS` query at the last layer of the transformer, create a binary mask where the $0.6\,N$ largest entries of this self-attention are set to 1, retrieve the largest spatially-connected component from this binary mask, and use the bounding box of this component as the discovered object. This method returns one box per self-attention head and we report results obtained with the best performing head over the entire dataset, noted as DINO-seg. LOST improves over DINO-seg by 8 to 17 of CorLoc points, demonstrating the efficacy of our approach for object localization based on self-supervised pre-trained transformer features.

Finally, we also evaluate our unsupervised class-agnostic detector (denoted by '+ CAD') for single-object discovery. To this end, we return for each image the box that the detector assigns the highest score. It can be seen that training a class-agnostic detector on LOST's outputs further improves the performance by 4 to 7 CorLoc points. In total, our method surpasses the prior state of the art by at least 10 CorLoc points on each evaluated dataset.

**Impact of the backbone architecture.** Table 5.2 studies the effect of the backbone on LOST. We see that transformer representations are better suited for our method (best results with ViT-S/16). In contrast, our performance using the DINO-pre-trained ResNet-50 is significantly lower. It indicates that the performance of our method is not only due to the contributions of self-supervision but also to the property and quality of the specific features we extract.

| Backbone | pre-training | VOC07_trainval | VOC12_trainval | C20K |
|---|---|---|---|---|
| VGG16 | supervised | 42.0 | 47.2 | 30.2 |
| ResNet50 | supervised | 33.5 | 39.1 | 25.5 |
| ResNet50 | DINO | 36.8 | 42.7 | 26.5 |
| ViT-S/8 | DINO | 55.5 | 57.0 | 49.5 |
| ViT-S/16 | DINO | **61.9** | **64.0** | **50.7** |
| ViT-B/16 | DINO | 60.1 | 63.3 | 50.0 |

Table 5.2 – Single-object discovery performance in CorLoc of LOST with features originating from different backbones: ViT [Dosovitskiy, 2021] small (ViT-S) and base (ViT-B) with patch size $P = 8$ or 16, ResNet50 [He, 2016] pre-trained following DINO [Caron, 2021], and VGG16 [Simonyan, 2015a] and ResNet50 trained in a fully-supervised fashion on Imagenet [Deng, 2009].

### 5.4.3 Unsupervised Object Detection

Here we explore the application of LOST in unsupervised object detection. To that end, we use LOST's pseudo-boxes to train a Faster R-CNN model [Ren, 2015b] on the datasets. We measure detection performance using the *Average Precision at IoU 0.5* metric (AP50), which is commonly used in the PASCAL detection benchmark. As Faster R-CNN backbone, we use a ResNet50 pre-trained with DINO self-supervision, thus making our training pipeline fully-unsupervised. We trained the Faster R-CNN models using the detectron2 [Wu, 2019] implementation (more details in Appendix D).

**Pseudo-labels.** To generate pseudo-labels for the class-aware detectors, we apply K-means clustering on DINO-ViT-S tokens using as many clusters as the number of different classes in the dataset. Since the cluster-based pseudo-labels are "anonymous", to evaluate the detection results we must map the clusters to the ground-truth classes. Following prior work in image clustering [Bautista, 2016; Asano, 2019; Ji, 2019], we use the Hungarian algorithm [Kuhn, 1955] for that. We stress that this matching is only for reporting evaluation results; we do not use any human labels during training.

**Unsupervised class-aware detection.** Table 5.3 provides results of unsupervised class-aware object detectors trained with LOST (entry 'LOST + OD') on VOC07 dataset. An illustration of the objects detected by LOST + OD on COCO are also shown in Figure 5.4. We are not aware of any prior work that addresses unsupervised object detection on real-world images of complex scenes, as those in PASCAL, that does not use extra modalities. We could not compare to [Afouras, 2020; Tian, 2021] as we focus on image-only benchmarks.

We see that, although fully-unsupervised, our method accurately detects several object classes. For example, detection performance for classes *aeroplane*, *bus*, *dog*, *horse* and *train* is more than 50.0 points, and for *cat* it reaches 72.2 points. Even more so, for some classes our method achieves better AP50 than the weakly-supervised methods WSDDN [Bilen, 2016] and PCL [Tang, 2018c], which require image-level human labels. Although the results are not entirely comparable due to backbone differences between our method and the weakly-supervised ones (self-supervised ResNet50 vs. supervised VGG16), they still demonstrate the efficacy of

our method in unsupervised object detection, which is an extremely hard and ill-posed task. We also compute AP of the pseudo-boxes generated on VOC07 test by our method (entry 'LOST') using their assigned cluster id as pseudo-labels. As clearly shown by the table, training the detector on pseudo-boxes leads to a significantly higher AP than just finding pseudo-boxes. Finally, switching LOST's pseudo-boxes with those of rOSD for training the detector (adding pseudo-labels to rOSD pseudo-boxes by clustering DINO features in exactly the same way as in our method) degrades the performance (entry 'rOSD + OD').

| Method | Sup. | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | moto | person | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [Bilen, 2016] | weak | 39.4 | 50.1 | 31.5 | 16.3 | 12.6 | 64.5 | 42.8 | 42.6 | 10.1 | 35.7 | 24.9 | 38.2 | 34.4 | 55.6 | 9.4 | 14.7 | 30.2 | 40.7 | 54.7 | 46.9 | 34.8 |
| [Tang, 2018c] | weak | 54.4 | 9.0 | 39.3 | 19.2 | 15.7 | 62.9 | 64.4 | 30.0 | 25.1 | 52.5 | 44.4 | 19.6 | 39.3 | 67.7 | 17.8 | 22.9 | 46.6 | 57.5 | 58.6 | 63.0 | 43.5 |
| rOSD + OD | none | 38.8 | 44.7 | 25.2 | 15.8 | 0.0 | 52.9 | 45.4 | 38.9 | 0.0 | 16.6 | 24.4 | 43.3 | 57.2 | 51.6 | 8.2 | 0.7 | 0.0 | 9.1 | 65.8 | 9.4 | 27.4 |
| LOST | none | 42.8 | 0.0 | 16.4 | 3.9 | 0.0 | 32.4 | 17.1 | 26.2 | 0.0 | 14.2 | 11.3 | 28.1 | 43.9 | 15.8 | 2.2 | 0.0 | 0.1 | 5.6 | 39.9 | 2.3 | 15.1 |
| LOST + OD | none | 57.4 | 0.0 | 40.0 | 19.3 | 0.0 | 53.4 | 41.2 | 72.2 | 0.2 | 24.0 | 28.1 | 55.0 | 57.2 | 25.0 | 8.3 | 1.1 | 0.9 | 21.0 | 61.4 | 5.6 | 28.6 |
| LOST + OD$^\dagger$ | none | 62.0 | 38.5 | 49.3 | 23.1 | 4.2 | 57.0 | 41.9 | 70.4 | 0.0 | 3.6 | 18.9 | 30.8 | 52.8 | 45.5 | 12.5 | 0.6 | 9.1 | 9.0 | 67.2 | 0.8 | **29.9** |

Table 5.3 – Object detection performance (AP50) on VOC07 test. LOST + OD and rOSD + OD are trained on VOC07 trainval. LOST + OD$^\dagger$ is trained on the union of VOC07 and VOC12 trainval sets.

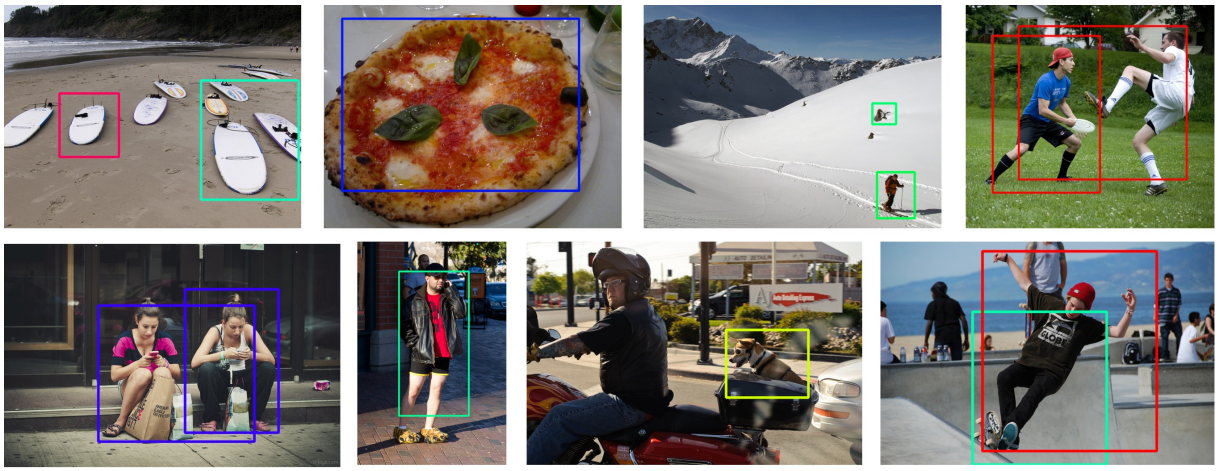


Figure 5.4 – Examples of predictions obtained by the class-aware detector LOST + OD on COCO (a different color per class). The actual "person" class is assigned three different pseudo-classes, illustrating the difficulty to "see" a single category for a "person" in very different configurations.

**Unsupervised class-agnostic detection.** We report in Table 5.4 class-agnostic detection results obtained using pseudo-boxes from our method ('LOST + CAD') as well as from rOSD ('rOSD + CAD') and LOD ('LOD + CAD'). Our method leads to significantly better detection performance. We also report detection results using the selective search [Uijlings, 2013] and EdgeBox [Zitnick, 2014] proposal algorithms, which perform worse than our method.

**Multi-object discovery results.** We compare in Table 5.5 the multi-object discovery performance of different methods. Following LOD, we use odAP50 and odAP[50:95] as evaluation metrics. As LOST only returns one region per image, we only consider here LOST + CAD,

| Training set (when applicable) | VOC07 trainval | | VOC12 trainval | C20K |
|---|---|---|---|---|
| Evaluation set | trainval | test | trainval | trainval |
| [Zitnick, 2014] | 3.6 | 4.4 | 4.8 | 1.8 |
| [Uijlings, 2013] | 2.9 | 3.6 | 4.2 | 1.6 |
| rOSD + CAD | 24.2 | 25.2 | 29.0 | 8.4 |
| LOD + CAD | 22.7 | 23.7 | 28.4 | 8.8 |
| LOST + CAD | **29.0** | **29.0** | **33.5** | **9.9** |

Table 5.4 – Class-agnostic unsupervised object detection performance in AP50. Trainings, corresponding to '*method* + CAD', are performed on the unlabelled images and rely only on the fully-unsupervised methods rOSD, LOD and LOST (ours). Evaluation of unsupervised object detection may thus be performed on the same images as those used for unsupervised training (without manual annotations). The classic methods EdgeBoxes [Zitnick, 2014] and Selective Search [Uijlings, 2013] do not involve any training.

which is the output of a class-agnostic detector (CAD) trained with LOST boxes, and we compare it to other approaches. It can be seen that LOST + CAD significantlyoutpe rforms all the previous methods, including the class-agnostic detector trained with LOD boxes (LOD + CAD).

| Method | odAP50 | | | odAP[50:95] | | |
|---|---|---|---|---|---|---|
| | VOC07_trainval | VOC12_trainval | C20K | VOC07_trainval | VOC12_trainval | C20K |
| [Kim, 2009] | 9.5 | 11.8 | 3.93 | 2.49 | 3.11 | 0.96 |
| [Wei, 2019] | 8.7 | 11.1 | 2.41 | 3.0 | 4.1 | 0.73 |
| rOSD | 13.1 | 15.4 | 5.18 | 4.29 | 5.27 | 1.62 |
| LOD | 13.9 | 16.1 | 6.63 | 4.47 | 5.34 | 1.98 |
| LOD + CAD | 15.8 | 20.9 | 7.26 | 5.03 | 7.07 | 2.28 |
| LOST + CAD | **19.8** | **24.9** | **7.93** | **6.71** | **8.85** | **2.51** |

Table 5.5 – Multi-object discovery performance in odAP (average precision for object discovery) of LOST and the baselines [Kim, 2009; Wei, 2019; rOSD; LOD].

**Image nearest neighbor retrieval.** Following LOD, we use LOST box descriptors to find images that are similar to each other (image neighbors) in the image collection. To this end, each image is represented by the `CLS` descriptors of its LOST box and the cosine similarity between these descriptors is used to define a similarity between the images. Then, for each image, the top $\tau$ images with the highest similarity are chosen as its neighbors. Similar to LOD, we choose $\tau = 10$ and use CorRet [Cho, 2015] as the evaluation metric, defined as the average percentage of the retrieved image neighbors that are actual neighbors (i.e., that contain objects of the same category) in the ground-truth image graph over all images. We compare the performance of our method in this task with rOSD and LOD in Table 5.6. We see that LOST boxes, when represented by DINO [Caron, 2021] features, yield a better CorRet score than the others. When VGG16 [Simonyan, 2015a] features are used, LOST is behind LOD but better than rOSD.

| Method | Features | CorRet (%) |
|---|---|---|
| rOSD | VGG16 | 64 |
| LOD | VGG16 | 70 |
| LOST | VGG16 | 68 |
| LOST | DINO | **72** |

Table 5.6 – Image neighbor retrieval performance (CorRet) of different methods and features.

### 5.4.4 Ablation Study

**Which transformer features to choose?** As explained in Section 5.3.2, we choose to use the keys $\mathbf{k}_p$ of the last attention layer as patch features $\mathbf{f}_p$ in LOST. As we will see here, this choice provides the best localization performance among the alternatives. Specifically, we report in the first section of Table 5.7 the performance of LOST when using as patch features $\mathbf{f}_p$ either the keys $\mathbf{k}_p$, the queries $\mathbf{q}_p$, or the values $\mathbf{v}_p$ of the attention layer. We see that the performance of LOST when using the queries $\mathbf{q}_p$ or the values $\mathbf{v}_p$ deteriorates by at least 11 CorLoc points compared to using the keys $\mathbf{k}_p$. Note that the better performance of keys compared to values or queries for localization tasks has also been observed in the more recent work of [Amir, 2021].

Another way to measure the similarity between two patches in a transformer architecture is to use the scalar product between the queries and the keys. We thus test substituting

$$\tilde{a}_{pq} = \begin{cases} 1 & \text{if } \mathbf{q}_p^\top \mathbf{k}_q + \mathbf{k}_p^\top \mathbf{q}_q \geq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{5.5}$$

for $a_{pq}$ in Equation (5.1) when selecting the initial seed. Note that this choice of $\tilde{a}_{pq}$ ensures the symmetry of the adjacency matrix. We test this new choice of similarity matrix when using the queries, keys or values in the seed expansion step, i.e., in $\mathcal{S}$, and in the box extraction steps, i.e., in $\mathbf{m}$ as defined in Equation (5.4).

Finally, we also try changing the definition of $\mathcal{S}$ to $\tilde{\mathcal{S}} = \{q \mid q \in \mathcal{D}_k \text{ and } \mathbf{q}_q^\top \mathbf{k}_{p^*} + \mathbf{k}_q^\top \mathbf{q}_{p^*} \geq 0\}$ and replacing the definition of $m_q$ in Equation (5.4) by

$$\tilde{m}_q = \begin{cases} 1 & \text{if } \sum_{s \in \mathcal{S}} \left( \mathbf{k}_q^\top \mathbf{q}_s + \mathbf{q}_q^\top \mathbf{k}_s \right) \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

Results in Table 5.7 show that all these alternatives using queries and keys yield results that are not as good as when using the keys as patch features.

**Importance of the seed expansion step.** We analyse here the importance of the seed expansion step that is controlled by $k$. The seed expansion step allows us to enlarge the region of interest so as to include all the parts of an object and not only the part localized from the initial seed. The last section of Table 5.7 presents the impact of the parameter $k$, which corresponds to the maximum number of patches that can be used to construct the mask $\mathbf{m}$. We notice that, without seed expansion (i.e., $k = 1$), there is a drastic drop in localization performance. The performance improves when increasing $k$ to 100-150 with a slight decrease at 200.

| Seed selection | Expansion & Box extrac. | $k$ | CorLoc |
|---|---|---|---|
| $a_{pq}$ with $\mathbf{f} = \mathbf{q}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{q}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 30.8 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{v}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{v}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 50.5 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{k}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 61.9 |
| $\tilde{a}_{pq}$ defined in Eq. (5.5) | $\mathbf{f}_p = \mathbf{q}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 30.8 |
| $\tilde{a}_{pq}$ defined in Eq. (5.5) | $\mathbf{f}_p = \mathbf{v}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 29.9 |
| $\tilde{a}_{pq}$ defined in Eq. (5.5) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 100 | 30.7 |
| $\tilde{a}_{pq}$ defined in Eq. (5.5) | using $\tilde{\mathcal{S}}$ and $\tilde{m}_q$ | 100 | 30.8 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{k}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 1 | 38.3 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{k}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 50 | 58.8 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{k}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 150 | 61.8 |
| $a_{pq}$ with $\mathbf{f} = \mathbf{k}$ in Eq. (5.1) | $\mathbf{f}_p = \mathbf{k}_p$ in $\mathcal{S}$ and $m_q$ | 200 | 61.2 |

Table 5.7 – CorLoc performance on VOC2007 with different choices of transformer features in the seed selection, expansion and box extraction steps, as well as influence on the results of the parameter $k$ (maximum number of patches with the lowest degree, in $\mathcal{D}_k$, for seed expansion).

Visualizations of results with $k = 1$ and $k = 100$ are presented in Figure 5.3 in Section 5.3.2 and Figure 5.5. We see that the boxes in yellow obtained with $k = 1$ are small and localized on probably what is the most discriminative part of the objects. Increasing $k$ permits us to increase the size of the box and localize the object better. We also present in Figure 5.6 cases of failure where the seed expansion step is either insufficient to localize the whole object or yields a box containing multiple objects.

**Impact of the number of clusters on class-aware detection training.** We assume in our unsupervised class-aware detection experiments that we know the exact number of object classes present in the used dataset, i.e., 20 in the VOC dataset, and use the same number of K-means clusters. Here we only assume that we have a rough estimate of the number of classes and study the impact of the number of clusters on the performance of the unsupervised detector. To this end, in Table 5.8, we provide the mean AP50 across all the 20 VOC classes when using 20, 25, 30 and 40 clusters. When we use more clusters than the 20 classes of the VOC dataset, Hungarian matching, which is used for reporting the AP50 results, maps to the VOC classes only the 20 best fitted clusters. Thus, when reporting the per-class AP results, we ignore the detections in these unmatched clusters since they have not been mapped to any ground-truth class. We observe in Table 5.8 that our unsupervised detector achieves good results for all the numbers of clusters. Interestingly, for 30 and 40 clusters there is a noticeable performance improvement. Similar findings have been observed on prior clustering work [Ji, 2019; Tian, 2021; Afouras, 2021].

**Impact of the non-determinism of the K-means clustering.** We investigate the impact of the randomness in the K-means clustering on the results of the object detectors. To that end, we repeat 4 times, using different random seeds, the unsupervised class-aware object detection experiment LOST + OD$^\dagger$ (Table 5.3). We obtain a standard deviation of 0.8 for the AP50 metric, which shows that the method is fairly insensitive to the randomness of the clustering



Figure 5.5 – An illustration of the effect of seed expansion on VOC07. In each image, the red square represents the seed $p^*$, the yellow box is obtained using only $p^*$, and the purple box is obtained using all the seeds $\mathcal{S}$ with $k = 100$.

method.

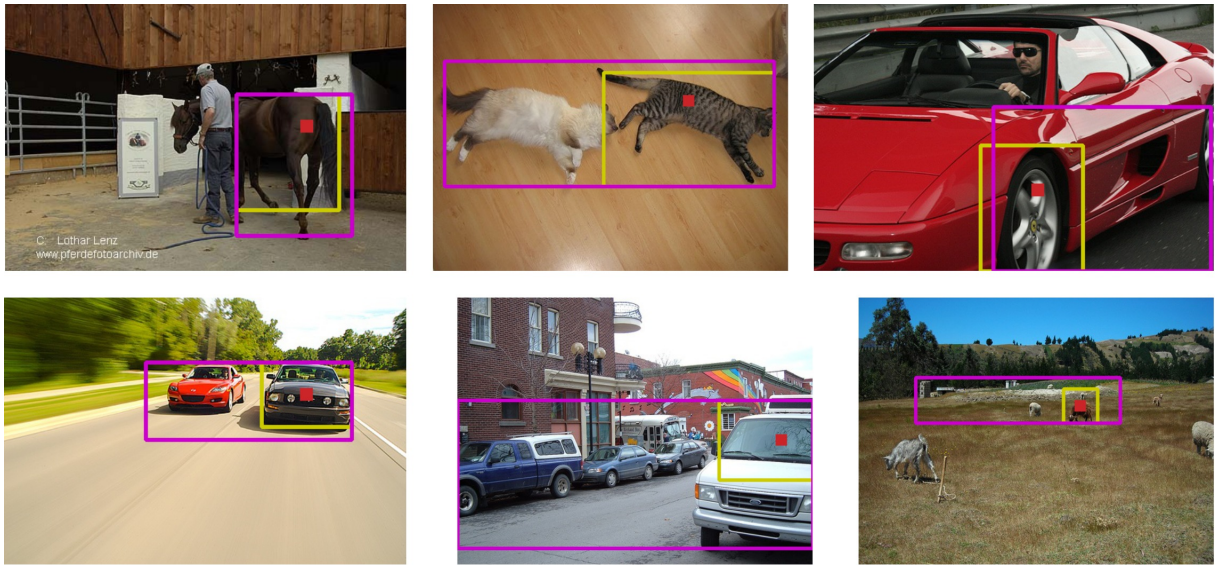



Figure 5.6 – Failure cases of seed expansion on VOC07. In each image, the red square represents the seed $p^*$, the yellow box is obtained using only $p^*$, and the purple box is obtained using all the seeds $\mathcal{S}$ with $k = 100$.

| Number $K$ of clusters | 20 | 25 | 30 | 40 |
|---|---|---|---|---|
| Mean AP (%) | 29.9 | 29.4 | 34.0 | 32.2 |

Table 5.8 – Impact of number of clusters in object detection. Results, using the mean AP50 (%) across all the classes, on VOC07 test. All models are trained using LOST's pseudo-boxes (i.e., LOST + OD) on the VOC07 and VOC12 trainval sets. The number of classes in VOC is 20.

## 5.5 Conclusion, Limitations and Future Work

We have presented LOST, a simple, yet effective method for localizing objects in images without any labels, by leveraging self-supervised pre-trained transformer features [Caron, 2021]. Despite its simplicity, LOST outperforms the state-of-the-art methods for object discovery by large margins. Having high precision, the boxes found by LOST can be used as pseudo ground truth for training a class-agnostic detector which further improves the object discovery performance. LOST boxes can also be used to train an unsupervised object detector that yields competitive results compared to weakly-supervised object detectors for several classes.

Despite the good performance of LOST, it exhibits of course some limitations. LOST, as it stands, can separate same-class instances that do not overlap (as it only keeps the connected component of the initial seed to create a box), but it is not designed to separate overlapping instances. This is actually a challenging problem, related to the difference between semantic [Long, 2015] and instance [He, 2017] segmentation tasks which, as far as we know, is an open problem in the absence of any supervision. A potential lead could be to use a matching algorithm such as probabilistic Hough matching [Cho, 2015] to separate instances within image regions found in multiple images. Another issue is when an object covers most of the image. It violates our second assumption for the initial seed selection (expressed in Section 5.3.2) that an

individual object covers less area than the background, thus possibly causing the seed to fall in the background instead of a foreground object. Ideally, we would like to filter out such failure cases, e.g., by using the attention maps of the `CLS` token.

Future work will be dedicated to addressing these limitations and investigating other applications of LOST boxes, *e.g.*, high-quality region proposals for object detection tasks, and the power of self-supervised transformer features for unsupervised object segmentation.

# Part II

# Annotation-efficient object detection

# Chapter 6

# Active Learning Strategies for Weakly-Supervised Object Detection

Objectives

Object detectors trained with weak annotations are affordable alternatives to their fully-supervised counterparts. However, there is still a significant performance gap between the two. We propose to narrow this gap by fine-tuning a base pre-trained weakly-supervised object detector with a few fully-annotated samples automatically selected from the training set using a novel active learning strategy named "box-in-box" (BiB) and designed specifically to address the well-documented failure modes of weakly-supervised detectors. Experiments on the VOC07 and COCO benchmarks show that BiB outperforms other active learning techniques and significantly improves the base weakly-supervised detector's performance with only a few fully-annotated images per class. BiB reaches 97% of the performance of fully-supervised Fast RCNN with only 10% of fully-annotated training images on VOC07. On COCO, using on average 10 fully-annotated images per class, or roughly 1% of the training set, BiB also reduces the performance gap (in AP) between the weakly-supervised detector and the fully-supervised Fast RCNN by over 70%, showing a good trade-off between performance and data efficiency.
This work, done in collaboration with Oriane Siméoni, Spyros Gidaris, Andrei Bursuc, Patrick Pérez and Jean Ponce, has appeared in Proceedings of the European Conference on Computer Vision (ECCV) 2022.

## Contents


6.1 Introduction . . . . . **96**
6.2 Proposed Approach . . . . . **97**
6.2.1 Problem Statement . . . . . 97
6.2.2 Active Learning for Weakly-Supervised Learning Object Detection . . . . . 98
6.2.3 BiB : An Active Learning Strategy . . . . . 98
6.2.4 Training Detectors with both Weak and Strong Supervision . . . . . 101
6.3 Experimental Analysis . . . . . **102**
6.3.1 Experimental Setup . . . . . 102
6.3.2 Experimental Results . . . . . 105
6.3.3 Additional Analysis . . . . . 109
6.4 Conclusion and Future Work . . . . . **110**

## 6.1 Introduction

Object detectors are critical components of visual perception systems deployed in real-world settings such as robotics or surveillance. Many methods have been developed to build object detectors with high predictive performance [Girshick, 2014; Gidaris, 2015; Girshick, 2015; Ren, 2015a; He, 2017] and fast inference [Redmon, 2016; Redmon, 2017]. They typically train a neural network in a fully-supervised manner on large datasets annotated manually with bounding boxes [Everingham, 2007; Everingham, 2012; Lin, 2014]. In practice, the construction of these datasets is a major bottleneck since it involves large, expensive and time-consuming data acquisition, selection and annotation campaigns. To address this challenge, much effort has been put in devising object detection approaches trained with less (or even no) human annotation. This includes semi-supervised [Radosavovic, 2018; Wang, 2018a; Jeong, 2019; Zoph, 2020; Li, 2020; Sohn, 2020b; Tang, 2021b; Xu, 2021], weakly-supervised [Bilen, 2016; Cinbis, 2017; Jie, 2017a; Tang, 2017; Tang, 2018a; Zeng, 2019; Arun, 2019; Gao, 2019b; Ren, 2020a; Huang, 2020], few-shot [Karlinsky, 2019; Kang, 2019; Fan, 2020; Sun, 2021], active [Settles, 2009; Geifman, 2017; Sener, 2018; Zhdanov, 2019; Brust, 2019; Agarwal, 2020; Haussmann, 2020; Yuan, 2021b; Choi, 2021] and unsupervised [Sivic, 2005; Russell, 2006; Tang, 2008; Kim, 2009; Cho, 2015; OSD; rOSD; LOD; LOST] learning frameworks for object detection.

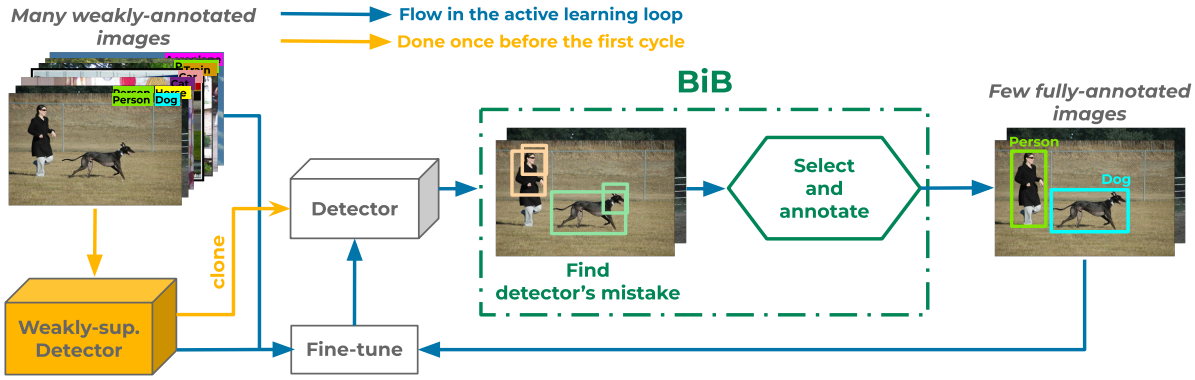



Figure 6.1 – Overview of our approach. A base weakly-supervised object detector is first trained with image-level tags only, then fine-tuned in successive stages using few *well-selected* images that are fully-annotated. For their selection, we propose *"box-in-box"* (BiB), an acquisition function designed to discover recurring failure cases of the weakly-supervised detector, e.g., failure to localize whole objects or to separate distinct instances of the same class.

Weakly-supervised object detection (WSOD) typically only uses image-level category labels during training [Bilen, 2016; Tang, 2017; Ren, 2020a]. This type of annotation is much cheaper than bounding boxes and, in some cases, it can even be obtained automatically, e.g., leveraging tags on online photos, photo captions in media or time-stamped movie scripts. WSOD is thus an affordable alternative to fully-supervised object detection in terms of annotation cost. However, weakly-supervised object detectors often struggle to correctly localize the full extent of objects [Tang, 2017; Ren, 2020a]. Several recent works [Pan, 2019; Biffi, 2020] show that a good trade-off between performance and annotation cost can be achieved by annotating objects in a small set of randomly selected training images with bounding boxes and training the detec-

tor with a mix of weak and full supervision. However, there are better alternatives to random selection. Active learning (AL) methods [Choi, 2021; Yuan, 2021b] *select* images that should be the most helpful for the training of an object detection model, given some criterion.

In this work, we propose to combine both worlds, by augmenting the weakly-supervised framework with an active learning scheme. Our active learning strategy specifically targets the known failure modes of weakly-supervised object detectors. We show that it can be used to significantly narrow the gap between weakly-supervised detectors and expensive fully-supervised ones with a few fully-annotated images per class.

We start with a weakly-annotated dataset, e.g., a set of images and their class labels, with which we train a weakly-supervised object detector. We apply a new active learning strategy that we call *box-in-box* (BiB) to iteratively select from the dataset a few images to be fully annotated at each stage and used to fine-tune the detector (Figure 6.1). Previous works have attempted to combine weak supervision with active learning, but they all start with an initial set of hundreds to thousands of fully-annotated images. As shown in Section 6.3, our approach only requires a very small number of fully-annotated images ($50-250$ on VOC07 [Everingham, 2007] and $160-800$ on COCO [Lin, 2014]) to significantly improve the performance of weakly-supervised detectors. Our main contributions are:

- We propose a new approach to improve weakly-supervised object detectors, by using a few fully-annotated images, carefully selected with the help of active learning. Contrary to typical active learning approaches, we initiate the learning process without any fully-annotated data.
- We introduce BiB, an active selection strategy that is tailored to address the limitations of weakly-supervised detectors.
- We validate our proposed approach with extensive experiments on VOC07 and COCO datasets. We show that BiB outperforms other active learning strategies on both datasets, and reduces significantly the performance gap between weakly- and fully-supervised object detectors.

## 6.2 Proposed Approach

### 6.2.1 Problem Statement

We assume that we are given $n$ images $\mathcal{I} = \{\mathbf{I}_i\}_{i\in\{1\dots n\}}$ annotated with labels $\mathcal{Q} = \{\mathbf{q}_i\}_{i\in\{1\dots n\}}$. Here $\mathbf{q}_i \in \{0,1\}^C$ is a vector encoding the class labels of image $\mathbf{I}_i$, with $C$ being the number of classes in the dataset. Let $M^0$ be a weakly-supervised object detector trained using only $\mathcal{Q}$. The goal of our work is to iteratively select a *very small set of images* to fully annotate with bounding boxes and fine-tune $M^0$ on the same images with both weak and full annotation so as to maximize its performance. To that end, we propose a novel *active learning* method properly adapted to the aforementioned problem setting.

### 6.2.2 Active Learning for Weakly-Supervised Learning Object Detection

As typical in active learning, our approach iterates over several cycles in which an acquisition function first uses the available object detector to select images that are subsequently annotated by a human with bounding boxes, before the detector is updated with this additional data (Algorithm 6.1).

**Algorithm 6.1:** WSOD with Active Learning.

**Input:** Set $\mathcal{I}$ of weakly-labelled images, set $\mathcal{Q}$ of weak annotations, number $T$ of cycles, budget $B$ per cycle.
**Result:** Detector $M^T$, bounding box annotations $\mathcal{G}^T$.

$M^0 \leftarrow \texttt{train}(\mathcal{I}, \mathcal{Q})$ ▷ weakly-supervised pre-training
**for** $t = 1$ *to* $T$ **do**
  $A^t \leftarrow \texttt{select}(W^{t-1}, M^{t-1}, \mathcal{I}, \mathcal{Q}, B)$ ▷ select a batch $A^t$ of $B$ images
  $\mathcal{G}^t \leftarrow \mathcal{G}^{t-1} \cup \texttt{label}(\mathcal{I}, A^t)$ ▷ annotate new selection
  $S^t \leftarrow S^{t-1} \cup A^t, \;\; W^t \leftarrow W^{t-1} \setminus A^t$ ▷ update the sets
  $M^t \leftarrow \texttt{fine-tune}(\mathcal{I}, \mathcal{Q}, \mathcal{G}^t, M^0)$ ▷ fine-tune the model
**end**

Let $W^t \subset \{1, \dots, n\}$ be the set of indices of images with class labels only, and $S^t \subset \{1, \dots, n\}$ the set with bounding-box annotations at the $t$-th active learning cycle. The active learning process starts with $W^0 = \{1, \dots, n\}$ and $S^0 = \varnothing$. At each cycle $t > 0$, the acquisition function selects from $W^{t-1}$ a set $A^t$ of $B$ images to be annotated with bounding boxes, with $B$ the fixed annotation budget per cycle. By definition, we have that $A^t \subset W^{t-1}$ and $|A^t| = B$. For the selection, the acquisition function exploits the detector $M^{t-1}$ obtained at the end of the previous cycle. After selecting $A^t$, the sets of fully and weakly-annotated images are updated with $S^t = S^{t-1} \cup A^t$ and $W^t = W^{t-1} \setminus A^t$ respectively. We define as $\mathcal{G}^t = \{\mathbf{G}_i\}_{i \in S^t}$ the bounding-box annotations for images with indices in $S^t$. Finally, at the end of cycle $t$, we fine-tune $M^0$ on the entire dataset, using the bounding-box annotations for images with indices in $S^t$ and the original image-level annotations for others.

### 6.2.3 BiB: An Active Learning Strategy

With a very small annotation budget, we aim at selecting the "best" training examples to "fix" the mistakes of the base weakly-supervised object detector. We propose BiB, an acquisition strategy tailored for this purpose. It first discovers (likely) detection mistakes of the weakly-supervised detector, and then selects diverse set of images containing those. Our selection strategy is summarized in Algorithm 6.2.

**Discovering BiB patterns.** Weakly-supervised object detectors often fail to recover the full extent of the objects in an image, and tend to focus instead on the most discriminative parts of an object or to group together multiple object instances [Ren, 2020a]. Several examples of these errors are shown in Figure 6.2. In the first column, a predicted box focuses on the most discriminative part of an object while a bigger one encompasses a much larger portion of the same

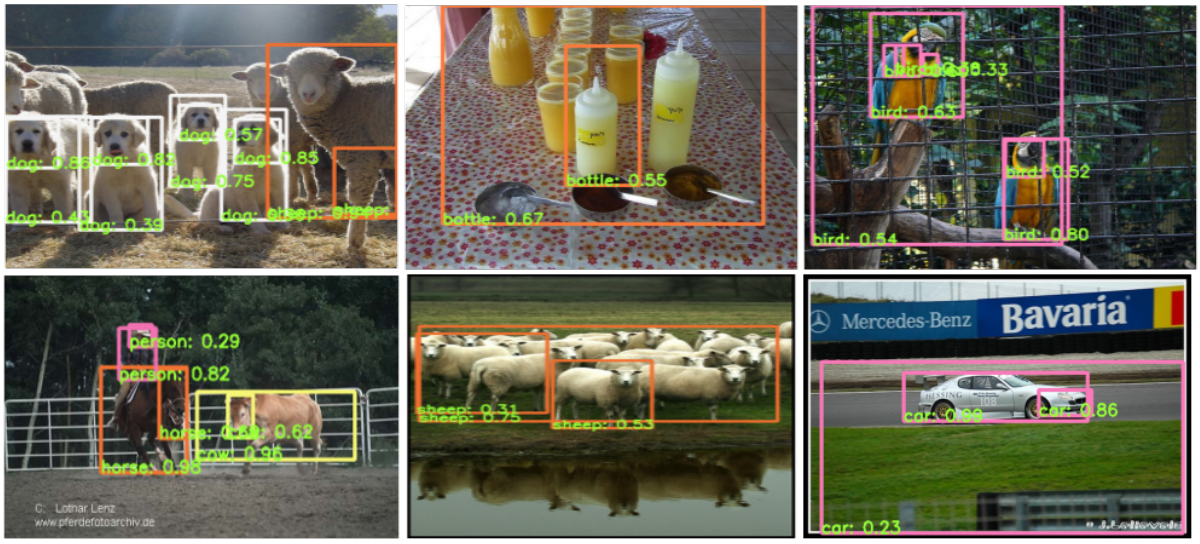



Figure 6.2 – Example of box-in-box (BiB) pairs among the predictions of the weakly-supervised object detector. The existence of such pairs is an indicator of the detector's failure on those images. In the images, boxes of different colors are prections of different classes and the numbers represent the prediction confidence. Best viewed in color.

object. Another recurring mistake is when two or more distinct objects are grouped together in a box, but some correct individual predictions are also provided for the same class (second column). The two kinds of mistakes can also be found in the same image (third column). We name "box-in-box" (BiB) such detection patterns where a pair of boxes is predicted for a same object class, the smaller box being "contained" (within some tolerance, see below) in the larger one. We take BiB pairs as an indicator of detector confusion.

Concretely, let $\mathbf{D}_i$ be the set of boxes detected in image $\mathbf{I}_i$ and let $d_A$ and $d_B$ be two of them. We consider that $(d_A, d_B)$ is a BiB pair, which we denote with is-bib$(d_A, d_B)$ = `True`, when: (i) $d_A$ and $d_B$ are predicted for the same class, (ii) $d_B$ is at least $\mu$ times larger than $d_A$ (i.e., $\frac{\text{Area}(d_B)}{\text{Area}(d_A)} \geq \mu$), and (iii) the intersection of $d_B$ and $d_A$ over the area of $d_A$ is at least $\delta$ (i.e., $\frac{\text{Intersection}(d_A, d_B)}{\text{Area}(d_A)} \geq \delta$). Hence, the set $P_i = \{p_{i,j}\}_{j=1}^{|P_i|}$ of BiB pairs is found in image $\mathbf{I}_i$ by the procedure

$$\text{find-bib}(\mathbf{D}_i) = \{(d_A, d_B) \in \mathbf{D}_i \times \mathbf{D}_i | \, \text{is-bib}(d_A, d_B)\}. \tag{6.1}$$

We observe that in such a BiB pair, it is likely that at least one of the boxes is a detection mistake (Section 6.3.3). We thus propose to select the images to be fully annotated among those containing BiB pairs.

**Selecting diverse detection mistakes.** Given the set of all BiB pairs over $\mathcal{I}$, the acquisition function considers the *diversity* of the pairs in order to select images. In particular, we follow k-means++ initialization [Arthur, 2007] – initially developed to provide a good initialization to k-means clustering by iteratively selecting as centroids data points that lie further away from the current set of selected ones. This initialization has previously been applied to image features

**Algorithm 6.2:** BiB acquisition strategy.

**Input:** Budget $B$, model $M^{t-1}$, image set $\mathcal{I}$, index set $W^{t-1}$ of weakly-annotated images, set $\hat{\mathcal{P}}$ of already selected BiB pairs (if empty, see text for initialization)
**Result:** Set $A^t$ of selected images.

```
for i ∈ W^{t-1} do
    D_i ← Detect(I_i | M^{t-1})                          ▷ Predict boxes
    P_i ← {p_{i,j}}_{j=1}^{|P_i|} = find-bib(D_i)        ▷ Discover BiB patterns
end
# Select diverse detection mistakes
A^t ← ∅
while |A^t| < B do
    for p ∈ ∪_{i∈W^{t-1}\A^t} P_i do
        w_p ← min_{p̃∈P̂} ||F(p) − F(p̃)||                 ▷ Comp. dist. to selected pairs
    end
    p* ∼ Prob({w_p}_p)                                   ▷ Randomly select a pair
    i* ← get-imid(p)                                     ▷ Get index of the image containing p*
    P̂ ← P̂ ∪ P_{i*}, A^t ← A^t ∪ {i*}                     ▷ Updates
end
```

in the context of active learning for object detection [Haussmann, 2020] or on model's gradients for active learning applied to image classification [Ash, 2020]. Here we focus and apply the algorithm to pairs of detected boxes.

We denote with $\hat{\mathcal{P}}$ the set of BiB pairs from the already selected images. For each pair $p$ not in $\hat{\mathcal{P}}$, we compute its minimum distance $w_p$ to the pairs in $\hat{\mathcal{P}}$: $w_p \leftarrow \min_{\tilde{p}\in\hat{\mathcal{P}}} \|F(p) - F(\tilde{p})\|$, where $F(p)$ is the feature vector associated with $p$, i.e., the concatenation of the region features corresponding to the two boxes in $p$ each extracted using the model $M^{t-1}$. We then randomly pick a new pair $p^*$, using a weighted probability distribution where a pair $p$ is chosen with probability proportional to $w_p$. We finally select the image $\mathbf{I}_{i^*}$ that contains $p^*$, add its index $i^*$ to $A^t$ and its BiB pairs to $\hat{\mathcal{P}}$. Note that at the beginning of the selection process in each cycle, $\hat{\mathcal{P}}$ contains the pairs of images selected in the previous cycles and is empty when the first cycle begins. In the latter case, we start by selecting the image $\mathbf{I}_{i^*}$ that has the greatest number of pairs $|P_{i^*}|$ [1] and add the pairs in $P_{i^*}$ to $\hat{\mathcal{P}}$ before starting the selection process above.

With this design, BiB selects a diverse set of images that are representative of the dataset while addressing the known mistakes of the weakly-supervised object detector. We show some examples selected by BiB and demonstrate its effectiveness in boosting the performance of the weakly-supervised detector in Section 6.3.2. The importance of selecting *diverse* BiB pairs is also discussed in Section 6.3.3.

### 6.2.4 Training Detectors with both Weak and Strong Supervision

We detail below how the model is fine-tuned each cycle. For clarity, we drop the image index $i$ and the cycle index $t$ in this section.

1. In case of a draw, an image is randomly selected.

**Training with weak annotations.** We adopt the state-of-the-art weakly-supervised method MIST [Ren, 2020a] as our base detector. MIST follows [Tang, 2017] which adapts the detection paradigm of Fast R-CNN [Girshick, 2015] to weak annotations. It leverages (pre-computed) region proposals extracted from unsupervised proposal algorithms, such as selective search [Uijlings, 2013] and EdgeBoxes [Zitnick, 2014]. In particular, given image $\mathbf{I}$ which has only weak labels $q$ (class labels) and its set of region proposals $\mathcal{R}$, simply called regions, the detection network extracts image features with a CNN backbone and computes for each region a feature vector using region-wise pooling [Girshick, 2015]. The network head(s) on top of the CNN backbone process the extracted region features in order to predict for each of them the object class and modified region coordinates. To build a detector that can be effectively trained using only image-wise labels, MIST has two learning stages, *coarse detection with multiple instance learning* and *detection refinement with pseudo-boxes*, each implemented with different heads but trained simultaneously in an online fashion [Tang, 2017].

The *multiple instance learner* (MIL) head is trained to minimize a multi-label classification risk $\mathcal{L}^{\text{MIL}}$ using weak labels and produce classification scores for all regions in $\mathcal{R}$. MIST selects the regions with the highest scores (with non-maximum suppression) as coarse predictions, which we denote with $\mathbf{D}^{(0)}$. These predictions are iteratively refined using $K$ consecutive *refinement heads*. Each refinement head $k \in \{1 \dots K\}$ predicts for all regions in $\mathcal{R}$ their classification scores for the $C+1$ classes ($C$ object classes plus 1 background class) and box coordinates per object class. The refinement head $k$ is trained by minimizing:

$$\mathcal{L}_w^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{D}^{(k-1)}) = \mathcal{L}_{\text{cls}}^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{D}^{(k-1)}) + \mathcal{L}_{\text{reg}}^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{D}^{(k-1)}), \tag{6.2}$$

which combines an adapted instance classification loss, $\mathcal{L}_{\text{cls}}^{(k)}$, and the box regression loss $\mathcal{L}_{\text{reg}}^{(k)}$ of Fast R-CNN [Girshick, 2015], using as targets the pseudo-boxes $\mathbf{D}^{(k-1)}$ generated by MIST from the region scores of the previous head. The final loss for image $\mathbf{I}$ is:

$$\mathcal{L}_w = \mathcal{L}^{\text{MIL}}(\mathbf{I}, \mathcal{R}, \mathbf{q}) + \sum_{k=1}^{K} \mathcal{L}_w^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{D}^{(k-1)}). \tag{6.3}$$

For more details about MIST, please refer to Appendix E and [Ren, 2020a].

**Adding strong annotations.** In our approach, we obtain ground-truth bounding boxes for *very few* images in the set $\mathcal{I}$. In order to integrate such strong annotations into the weakly-supervised framework, we simply replace the pseudo-annotations in Equation (6.2) with box annotations $\mathbf{G}$, now available for image $\mathbf{I}$. The resulting loss for the refinement head $k$ reads:

$$\mathcal{L}_s^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{G}) = \mathcal{L}_{\text{cls}}^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{G}) + \mathcal{L}_{\text{reg}}^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{G}), \tag{6.4}$$

and the final loss on image $\mathbf{I}$ in this case is:

$$\mathcal{L}_s = \mathcal{L}^{\text{MIL}}(\mathbf{I}, \mathcal{R}, \mathbf{q}) + \sum_{k=1}^{K} \mathcal{L}_s^{(k)}(\mathbf{I}, \mathcal{R}, \mathbf{G}). \tag{6.5}$$

While fine tuning the detector $M^0$, we use $\mathcal{L}_w$ for images for which only class labels are available and $\mathcal{L}_s$ for images provided with bounding boxes.

**Difficulty-aware proposal sampling.** In this framework, we use thousands of pre-computed proposals in $\mathcal{R}$ for each image. This is necessary when no box annotations are provided. However, when ground-truth boxes are available, better and faster training can be achieved by sampling a smaller number of proposals thanks to a stable ratio between negative and positive proposals [Girshick, 2015; Ren, 2020b]. In particular, we select a subset of 512 proposals with 25% of *positive* boxes, i.e., those whose IoU with one of the ground-truth boxes exceeds 0.5, and 75% of *negative* boxes, i.e., those whose IoU with all ground-truth boxes is smaller than 0.3. We have noticed that negative proposals are often over-sampled from the background or appear uninformative. We thus propose to improve negative proposal sampling by using the network predictions to select those classified as objects. First, a forward pass is performed and the average classification scores over the $K$ refinement heads is computed; we then apply row-wise softmax and select proposals with the highest class scores, excluding background. We show in our experiments that this negative proposal sampling method allows better training and yields better performance.

## 6.3 Experimental Analysis

In this section, we first introduce the general setup of our experiments. We then present an ablation study to assess the effectiveness of different components of BiB before comparing BiB to different existing active learning strategies. Finally, we compare our method to the state of the art.

### 6.3.1 Experimental Setup

**Datasets.** We evaluate our method on two well-known datasets for object detection, Pascal VOC2007 [Everingham, 2007] (noted VOC07) and COCO2014 [Lin, 2014] (COCO). Following previous works [Biffi, 2020; Ren, 2020a], we use the *trainval* split of VOC07 for training and the *test* split for evaluation. They contain 5011 and 4952 images respectively. On COCO, we train detectors with the *train* split (82783 images) and evaluate on the *validation* split (40504 images) following [Biffi, 2020]. We use the average precision metrics AP50 and AP, computed respectively with an IoU threshold of 0.5 and with thresholds in $[0.5 : 0.95]$. We report results corresponding to $N$-shot experiments – where $N \times C$ images are selected – and $N$% experiments, where about $N$ percents of the training set are selected to be fully-annotated.

**Architecture.** Though BiB can be applied on any weakly-supervised object detector, we use MIST [Ren, 2020a] as our base weakly-supervised detector for it has public code and has been shown to be a strong baseline. We modify MIST to account for images containing bounding-box annotations during training as detailed in Section 6.2.4. Concrete drop block (CDB), the spatial dropout technique introduced in [Ren, 2020a] to encourage the detector to focus more on

context instead of the most discriminative object parts, is used in MIST in our experiments on VOC07 but dropped in COCO experiments to save computational cost. We use our difficulty-aware proposal sampling in all experiments unless stated otherwise. We train with a batch size of 8 and a learning rate of 4e−4 for MIST and 4e−6 for CDB when the latter is used. During training, images are drawn from the sets of images with weak and strong annotation uniformly at random such that the numbers of weakly- and fully-annotated images considered are asymptotically equal.

**Active learning setup.** We emulate an active learning setup by ignoring available bounding box annotations of images considered weakly annotated in our experiments. On both datasets, we run MIST [Ren, 2020a] three times to account for the training's instability and obtain three base weakly-supervised object detectors. We fine-tune each base weakly-supervised detector twice on VOC07 and once on COCO, giving respectively 6 and 3 repetitions. We always report the averaged results of these repetitions, and in some cases also the standard deviation. The number of fine-tuning iterations is scaled linearly with the number of fully-annotated images in the experiment. Concretely, the base weakly-supervised detector is fine-tuned over 300 iterations for every 50 fully-annotated images in VOC07 and 1200 iterations for every 160 images on COCO. If not mentioned otherwise, we set $\mu = 3$ and $\delta = 0.8$ in BiB. We provide a study on their influence in Section 6.3.3.

**Active learning baselines.** We compare BiB to different active learning strategies in our experiments. We provide here details about them. As described in Algorithm 6.1, a set of images $A^t$ of $B$ images is selected at each cycle $t$. The selection is performed with an active learning method within the set of images $W^{t-1}$, possibly using the detector $M^{t-1}$ trained at the end of the previous cycle and the set of selected images $S^{t-1}$.

*Random.* We implement two variants of the random sampling: *u-random* and *b-random*. In *u-random*, $B$ images are selected uniformly at random from $W^{t-1}$; *b-random* seeks to have a balance sampling among the classes. Images are iteratively selected until the budget $B$ is reached. At each iteration, an image containing at least an object of the class that is the least represented [2] in $S^{t-1} \cup A^t$ is randomly chosen and added to $A^t$.

*Diversity-based strategies.* The core-set [Sener, 2018] approach attempts to select a representative subset of a dataset. We employ the greedy version of *core-set* in our experiments. In particular, at cycle $t$, let $\psi_{t-1}(\mathbf{I}_i)$ be the features of image $\mathbf{I}_i$ extracted from detector $M^{t-1}$, *core-set* iteratively selects the image $i^*$ to be added in $A^t$ by solving the optimization problem:

$$i^* = \underset{i \in W^{t-1} \setminus A^t}{\operatorname{argmax}} \min_{j \in S \cup A^t} \|\psi_{t-1}(\mathbf{I}_i) - \psi_{t-1}(\mathbf{I}_j)\|. \tag{6.6}$$

In the first cycle, the very first image is randomly selected.

*Selection using model uncertainty.* The concept of informativeness has been widely exploited in the literature [Yoo, 2019; Brust, 2019; Choi, 2021; Yuan, 2021b]. For a classification task, the

2. In case of draw, a class is randomly selected.

uncertainty can be computed by measuring the entropy over the class predictions of an image. Here, we first compute the entropy over the class predictions of each predicted box in an image, and then the box-entropy scores of an image are aggregated using the *sum* and *max* pooling, resulting in two strategies, *entropy-sum* and *entropy-max*. Concretely, let $p_{i,j} \in \mathbb{R}^{C+1}$ be the predicted class scores of the predicted box $j$ for image $\mathbf{I}_i$ given by $M^{t-1}$, and $\mathbf{D}_i$ be the set of all predictions in $\mathbf{I}_i$, we compute the uncertainty score $u_i$ of image $\mathbf{I}_i$ as

$$u_i = \max_{1 \leq j \leq |\mathbf{D}_i|} \sum_{c=1}^{C+1} -p_{i,j}^c \log(p_{i,j}^c) \tag{6.7}$$

for *entropy-max* and

$$\sum_{1 \leq j \leq |\mathbf{D}_i|} \sum_{c=1}^{C+1} -p_{i,j}^c \log(p_{i,j}^c) \tag{6.8}$$

for *entropy-sum*. Then, the $B$ images with the highest scores $u_i$ are selected.

*Combining diversity and uncertainty.* Following [Haussmann, 2020], we consider a selection strategy function that incorporates the uncertainty information into *core-set* by multiplying the distances between image features with the uncertainty score $u_i$ defined above. Specifically we combine *core-set* and *entropy-max*, in a new active learning method *core-set-ent* which iteratively selects an image $i^*$ following:

$$i^* = \operatorname*{argmax}_{i \in W^{t-1} \setminus A^t} \min_{j \in S \cup A^t} u_i \times \|\psi_{t-1}(\mathbf{I}_i) - \psi_{t-1}(\mathbf{I}_j)\|. \tag{6.9}$$

*Selection using losses.* [Yoo, 2019] proposes to learn, through an auxiliary module, an object detection loss predictor which later allows choosing samples that produce the highest losses. Conveniently, the refinement heads of MIST produce refinement losses ($\mathcal{L}_w^{(k)}$ with $k \in \{1, 2, 3\}$) that are detection losses computed using pseudo-boxes. We therefore propose the active learning method *loss* which selects the $B$ images with the highest loss $\mathcal{L}_w^{(3)}$, which yields the best results amongst the losses produced by different refinement heads of MIST. We provide results obtained when considering other losses in Appendix E.

### 6.3.2 Experimental Results

**Ablation studies.** We perform in Table 6.1 an ablation study to understand the relative importance of the difficulty-aware proposal sampling (*DifS*), the selection based on k-means++ initialization, and the use of box-in-box pairs in our method. The second row corresponds to *u-random*. We apply the diversity selection (e.g., following k-means++ initialization) on image-level features, predictions, and BiB pairs. The experiments are conducted on VOC07, and for each variant of our method, we perform 5 active learning cycles with a budget of 50 images per cycle. It appears that *DifS* significantly improves results over both random and BiB selection, confirming that targeting the detector's most confusing regions is helpful. K-means++ initialization does not help when applied on image-level features but yields significant performance boosts over random when combined with region-level features. Finally, the use of

BiB pairs shows consistent improvements over *region*, confirming our choices in BiB's design.

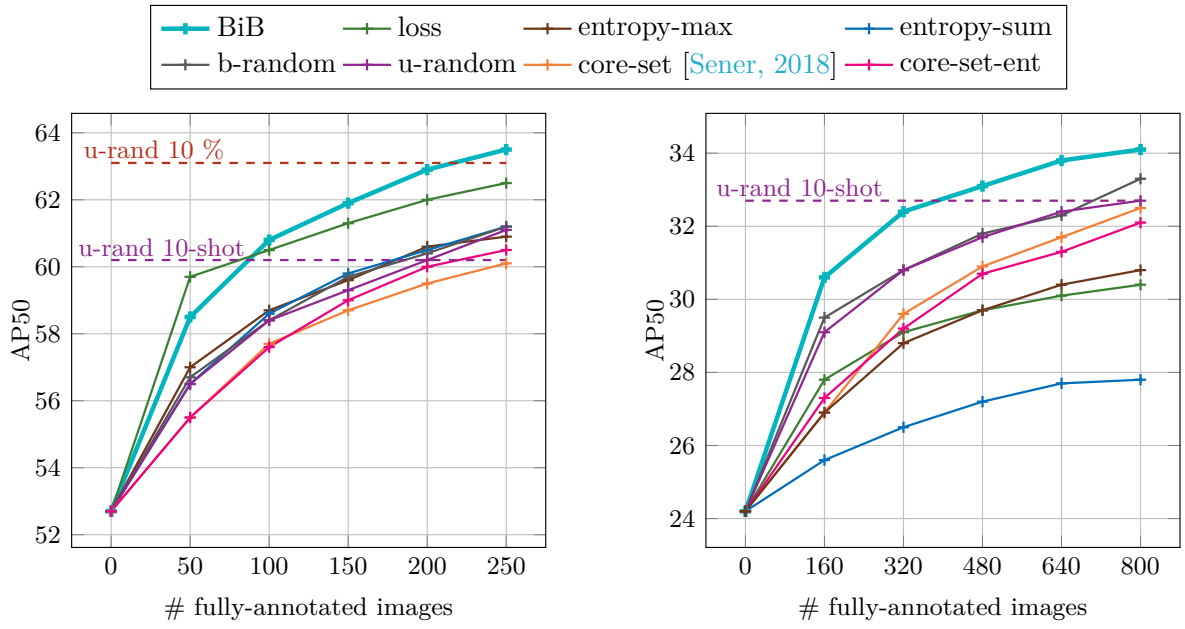
| DifS | K selection | | | Number of images annotated | | | | |
|---|---|---|---|---|---|---|---|---|
| | im. | reg. | BiB | 50 | 100 | 150 | 200 | 250 |
| | | | | 56.3 ± 0.4 | 58.0 ± 0.5 | 58.9 ± 0.4 | 60.0 ± 0.3 | 60.5 ± 0.4 |
| ✓ | | | | 56.5 ± 0.4 | 58.4 ± 0.4 | 59.3 ± 0.7 | 60.2 ± 0.4 | 61.1 ± 0.5 |
| ✓ | ✓ | | | 57.1 ± 0.4 | 58.3 ± 0.5 | 59.3 ± 0.6 | 59.8 ± 0.4 | 60.3 ± 0.4 |
| ✓ | | ✓ | | **58.4 ± 0.4** | 60.2 ± 0.4 | 61.5 ± 0.6 | 62.6 ± 0.4 | **63.4 ± 0.3** |
| | | | ✓ | 57.9 ± 0.7 | 60.1 ± 0.4 | 61.2 ± 0.5 | 62.1 ± 0.5 | 62.6 ± 0.4 |
| ✓ | | | ✓ | **58.5 ± 0.8** | **60.8 ± 0.5** | **61.9 ± 0.4** | **62.9 ± 0.5** | **63.5 ± 0.4** |

Table 6.1 – Ablation study. Results in AP50 on VOC07 with 5 cycles and a budget $B = 50$. We provide averages and standard deviation results over 6 repetition. *DifS* stands for the difficulty-aware region sampling module. Images are selected by applying k-means++ init. (*K selection*) on image-level features (*im.*), confident predictions' features (*reg.*) or BiB pairs.

**Comparison of active strategies.** In order to compare BiB to baselines, we conduct 5 active learning cycles with a budget of $B = 50$ images (1% of the training set) per cycle on VOC07 and of $B = 160$ images (0.2% of the training set, 2 fully-annotated images per class on average) on COCO. We present results in Figure 6.3. The detailed numbers are provided in Appendix E. It can be seen that the ranking of the examined baseline methods based on their detection performance is different on the two datasets. This is explained by the fact that the two datasets have different data statistics. COCO dataset contains many cluttered images, with an average of 7.4 objects in an image, and VOC07 depicts simpler scenes, with an average of only 2.4 objects. However, BiB consistently improves over other baselines.

Results on VOC07 (Figure 6.3a) show that BiB and *loss* significantly outperform every method in all cycles. BiB also surpasses *loss* except in the first cycle. Entropy and variants of *random* perform comparably and slightly better than variants of core-set. Balancing the classes consistently improves the performance of random strategy, albeit by a small margin. Interestingly, BiB reaches the performance of *random* at 10% setting ($\approx 500$ images) with only about 200 fully-annotated images. Similarly, it needs fewer than 100 fully-annotated images to attain *random*'s performance in the 10-shot ($\approx 200$ images) setting.

On COCO, BiB again shows consistent improvement over competitors. However, surprisingly, *loss* fares much worse than BiB and even *random*. To understand these results, we present a representative subset of selected images in Figure 6.4. It appears that images selected by the *loss* strategy tend to depict complex scenes. Many of them are indoors scenes with lots of objects (people, food, furnitures, ...). The supervision brought by these images is both redundant (too many images for certain classes) and insufficient (no or too few images for others). This result agrees with those obtained in [Choi, 2021; Liu, 2021c] on COCO with the predicted loss method [Yoo, 2019]. On the other hand, variants of entropy strategy tend to select very difficult images that are not representative of the training dataset. They do not perform well on COCO, especially *entropy-sum* which obtains significantly worse results than other strategies. This observation is similar to that of [Yuan, 2021b]. Diversity-based methods fare better




(a) Results on VOC07.

(b) Results on COCO.

Figure 6.3 – Detection performances in AP50 of different active learning strategies in our framework on VOC07 [Everingham, 2007] (a) and COCO datasets [Lin, 2014] (b). We perform 5 annotation cycles for each strategy with the budget of $B = 50$ images per cycle on VOC07 and $B = 160$ images per cycle on COCO. This corresponds to annotating 1% and 0.2% of the training set per cycle respectively for the two datasets. Dashed lines in purple and red highlight results obtained with *10-shot* and *10%* images selected with *u-random*. Best viewed in color.

than uncertainty-based methods, with *core-set* and *core-set-ent* performing much better than *entropy* variants. Among the latter two methods, *core-set* performs unsurprisingly better than *core-set-ent*, given *entropy*'s bad performance. BiB outperforms all other methods. It obtains significantly better results than *random*, which other methods fail to do. In addition, BiB attains the same performance as *u-random* (see dashed line) with only half as many annotated images, reducing the performance gap (in AP50) between the base weak detector and the fully-supervised Fast RCNN by nearly 70% with only ten fully-annotated images per class on average. It can be seen in Figure 6.4 that BiB selects a diverse set of images that reflect the detector's confusion on object extent.

**Comparison to the state of the art.** We compare the 10-shot performance of our method to the state of the art in Table 6.2. For BiB, we report the performance obtained in the previous experiments (Figure 6.3) at cycle 4 on VOC07 and cycle 5 on COCO. All compared methods use a Fast R-CNN [Girshick, 2015] or Faster R-CNN [Ren, 2015a] architecture with a VGG16 [Simonyan, 2015b] backbone. Most related to us, OAM [Biffi, 2020] and BCNet [Pan, 2019] also seek to improve the performance of weakly-supervised object detectors with a few fully-annotated images. We can see that BiB significantly outperforms them in this setting. In particular, on COCO, we observe from Table 6.2 and Figure 6.3 that BiB obtains comparable results to 10-shot OAM with only 2 shots (160 images) and significantly better results with 4




Figure 6.4 – Images selected by BiB, *entropy-max* and *loss* strategies on COCO dataset. Images selected by *loss* tend to depict complex scenes, many of which are indoors scenes with lots of objects (people, food, furnitures, ...). The supervision brought by these images is both redundant (too many images for certain classes) and insufficient (no or too few images for others). *entropy-max* tends to select very difficult images that are not representative of the training dataset. In contrast, BiB selects a diverse set of images that reflect the detector's confusion on object extent. As a result, BiB significantly outperforms the others on this dataset.

shots. Similarly, on VOC07, BiB surpasses the performance of OAM with only a half of the number of fully-annotated images used by the latter.

We also consider the 10% setting and compare BiB to other baselines on the VOC07 dataset (see Table 6.3). In this setting, a random selection following our method ("Ours (u-rand)") gives an AP50 of 63.1, outperformed by BiB ("Ours (BiB)") which achieves an AP50 of 65.1. In comparison, our main competitors perform worse: OAM (63.3), BCNet (61.8), EHSOD [Fang, 2020] (55.3) and BAOD [Pardo, 2021] (50.9).

Compared to WSOD methods, we obtain significantly better results with a small amount of full annotations. BiB enables a greater boost over weakly-supervised detectors than *random* and significantly narrows the performance gap between weakly-supervised and fully-supervised detectors. It reduces the gap between the state-of-the-art weakly-supervised object detector CASD [Huang, 2020] and Fast RCNN [Girshick, 2015] by 5.5 times with 10% of the training images fully annotated on VOC07 and by 3.5 times with only 10 fully-annotated images on average per class on COCO. This is arguably a better trade-off between detection performance and data efficiency than both weakly- and fully-supervised detection.

**Per-class study.** Additionally, we present in Table 6.3 the per-class results for different methods on VOC07. It can be seen that variants of our approach (*u-random* and BiB) consistently boost the detection performance on all classes over MIST [Ren, 2020a] (except on *aeroplane* and *motorbike* where they perform slightly worse than MIST). Notably, BiB yields larger boosts on *hard* classes such as *table* (+23 points w.r.t. our baseline MIST), *chair* (+17.3), *bottle* (+23) and *potted plant* (+19.2). On those classes, a random selection with our approach is worse than BiB by more than 7 points. Overall, BiB obtains the best results on most classes.

| Setting | Method | VOC07 | COCO | |
|---|---|---|---|---|
| | | AP50 | AP50 | AP |
| | | 100% | | |
| Fully supervised | [Girshick, 2015] | 66.9 | 38.6 | 18.9 |
| | [Ren, 2015a] | 69.9 | 41.5 | 21.2 |
| | | 0% | | |
| WSOD | [Bilen, 2016] | 34.8 | - | - |
| | [Tang, 2017] | 41.2 | - | - |
| | [Gao, 2019b] | 52.6 | 21.4 | 9.6 |
| | [Zeng, 2019] | 53.6 | 22.7 | 10.8 |
| | [Ren, 2020a] | 54.9 | 24.3 | 11.4 |
| | [Huang, 2020] | 56.8 | 26.4 | 12.8 |
| | | 10-shot | | |
| Weak & few strong | [Pan, 2019] | 57.1 | - | - |
| | [Biffi, 2020] | 59.7 | 31.2 | 14.9 |
| | Ours (u-rand) | 60.2 | 32.7 | 16.4 |
| | Ours (BiB) | 62.9 | 34.1 | 17.2 |

Table 6.2 – Performance of BiB compared to the state of the art on VOC07 ($B = 50$) and COCO ($B = 160$) datasets. The *10-shot* setting corresponds to 4 and 5 AL cycles resp. on VOC07 and COCO. All of the compared methods use VGG16 [Simonyan, 2015b] as the backbone.

| Method | sup. | aero | bike | bird | boat | bottl | bus | car | cat | chair | cow | table | dog | horse | moto | pers | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [Ren, 2020a]* | ✗ | 69.0 | 75.6 | 57.4 | 22.5 | 24.8 | 71.5 | 76.1 | 55.9 | 27.6 | 70.3 | 43.9 | 37.5 | 50.8 | **75.9** | 18.5 | 23.9 | 60.8 | 54.7 | 69.3 | 68.1 | 52.7 |
| [Pardo, 2021] | 10% | 51.6 | 50.7 | 52.6 | 41.7 | 36.0 | 52.9 | 63.7 | 69.7 | 34.4 | 65.4 | 22.1 | 66.1 | 63.9 | 53.5 | 59.8 | 24.5 | 60.2 | 43.3 | 59.7 | 46.0 | 50.9 |
| [Pan, 2019] | 10% | 64.7 | 73.1 | 55.2 | 37.0 | 39.1 | **73.3** | 74.0 | 75.4 | 35.9 | 69.8 | 56.3 | **74.7** | 77.6 | 71.6 | 66.9 | 25.4 | 61.0 | 61.4 | **73.8** | 69.3 | 61.8 |
| [Biffi, 2020] | 10% | 65.6 | 73.1 | 59.0 | **49.4** | 42.5 | 72.5 | 78.3 | **76.4** | 35.4 | 72.3 | 57.6 | 73.6 | **80.0** | 72.5 | **71.1** | 28.3 | 64.6 | 55.3 | 71.4 | 66.2 | 63.3 |
| Ours (u-r.) | 10% | **70.5** | 77.2 | 62.3 | 38.5 | 38.5 | 72.3 | **79.4** | 73.6 | 38.6 | 73.8 | 55.7 | 66.5 | 71.4 | 75.3 | 65.5 | 33.8 | 65.4 | 62.7 | 72.3 | 69.7 | 63.1 |
| Ours (BiB) | 10% | 68.9 | **78.1** | **62.7** | 41.4 | **47.8** | 72.4 | **79.2** | 70.3 | **44.9** | **74.7** | **66.2** | 62.2 | 72.1 | **75.6** | 69.8 | **43.1** | **66.2** | **65.0** | 71.4 | **70.7** | **65.1** |

Table 6.3 – Per-class AP50 results on VOC07. BiB yields significant boosts in hard classes such as *bottle*, *chair*, *table* and *potted plant*. Results of MIST [Ren, 2020a] are the average of three runs using the authors' public code and differ from the numbers in the original paper.

**Qualitative results.** We show in Figure 6.5 predictions obtained with the weakly-supervised object detector MIST (top row) and the detector after the first cycle of BiB (bottom row) with $B = 50$ on VOC07 and $B = 160$ on COCO. We observe that the failure modes of MIST are corrected in this case by our BiB detector: objects and parts are not confused ($3^{rd}$ and $4^{th}$ images), objects are covered ($1^{st}$ image) and better separated ($2^{nd}$ image).

### 6.3.3 Additional Analysis

**Are diverse samples important?** We propose in BiB to find diverse images on which the weakly-supervised object detector fails. We investigate here the importance of sample diversity in BiB by comparing it to two variants. In the first variant, we randomly select images containing BiB pairs ('U(BiB)') and, in the second variant, we use a mix, with half selected with BiB and the other half with randomly uniform sampling ('U+BiB'), to be fully annotated. We show the results in Table 6.4. The fact that U(BiB) is worse than BiB and U+BiB outperforms U(BiB)




Figure 6.5 – Examples of predictions on the VOC07 and COCO test sets, by MIST [Ren, 2020a] (first row) and BiB after the first cycle (second row). Fine-tuning MIST with images selected by BiB significantly remedies its limitations.

in general shows that diversity sampling is important once BiB patterns have been discovered.

| Method | Dataset | AL cycles | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| u-rand. | VOC | 56.5 | 58.4 | 59.3 | 60.2 | 61.1 |
| U(BiB) | | 57.6 | 59.2 | 60.1 | 61.2 | 61.8 |
| U+BiB | | 57.9 | 59.4 | 60.7 | 61.6 | 62.4 |
| BiB | | **58.5** | **60.8** | **61.9** | **62.9** | **63.5** |
| u-rand | COCO | 29.1 | 30.8 | 31.7 | 32.4 | 33.0 |
| U(BiB) | | 30.0 | 31.4 | 32.3 | 33.1 | 33.5 |
| U+BiB | | 29.7 | 31.4 | 32.4 | 33.2 | 33.7 |
| BiB | | **30.6** | **32.4** | **33.1** | **33.8** | **34.1** |

Table 6.4 – A comparison between BiB, u-rand and two other variants that combine them. BiB outperforms the variants, showing that diversity sampling is important to the effectiveness of BiB.

**Verification of BiB pairs.** BiB pairs are used in this work as an indicator of a detector's confusion on images. With this design, we argue that at least one box in the pair is likely a wrong prediction. We verify this assumption on MIST's predictions on VOC07 and COCO. Among 8,758 BiB pairs on VOC, there are 8,633 pairs (98.6%) with at least one wrong prediction while 99.6% of the 854,004 BiB pairs have at least one wrong box on COCO.

**Number of BiB pairs reduced with active learning cycles.** Intuitively, as the model becomes more accurate with more active learning cycles, fewer BiB pairs should be found. We have computed the number of BiB pairs during active learning cycles on VOC07 and COCO datasets to verify this assumption. As expected, our results show that this number decreases with iterations. On VOC, it drops from 8801 in cycle 1 to 5170 in cycle 5 with budget $B = 50$. On COCO, it decreases from 854k in cycle 1 to 152k in cycle 5 with budget $B = 160$.

**Influence of hyper-parameters.** We use two intuitive hyper-parameters in BiB: the area ratio $\mu$ between two boxes in a BiB pair and the ratio $\delta$ of the overlap over the smallest box. By design, the latter should be close to 1 so that the small box is "contained" in the large box, and it is set to 0.8 in our experiments. For the former, we test BiB on VOC07 when its value varies in $\{2, 3, 4\}$ and report results in Table 6.5. It can be seen that the performance is relatively insensitive to $\mu$. We use $\mu = 3$ in our experiments.

| | Number of fully-annotated images | | | | |
|---|---|---|---|---|---|
| $\mu$ | 50 | 100 | 150 | 200 | 250 |
| $\mu = 2$ | 58.5 $\pm$ 0.5 | 60.4 $\pm$ 0.3 | 61.6 $\pm$ 0.4 | 62.4 $\pm$ 0.3 | 63.1 $\pm$ 0.2 |
| $\mu = 3$ | 58.5 $\pm$ 0.8 | 60.8 $\pm$ 0.5 | 61.9 $\pm$ 0.4 | 62.9 $\pm$ 0.5 | 63.5 $\pm$ 0.4 |
| $\mu = 4$ | 58.3 $\pm$ 0.5 | 60.6 $\pm$ 0.3 | 61.7 $\pm$ 0.3 | 62.5 $\pm$ 0.4 | 63.3 $\pm$ 0.2 |

Table 6.5 – Performance of BiB on VOC07 with different values of the area ratio $\mu$ in BiB design. We conducted 5 cycles with a budget of 50 images per cycle, repeated the experiment six times for each value of $\mu$ and report the average and standard deviation of their performance.

## 6.4 Conclusion and Future Work

We have proposed a new approach to boost the performance of weakly-supervised object detectors using a few fully-annotated images selected following an active learning process. We introduce BiB, a new selection method specifically designed to tackle failure modes of weakly-supervised detectors and show a significant improvements over random sampling – BiB requires less than half the data to achieve the same results. Moreover, BiB is effective on both VOC07 and COCO datasets, narrowing significantly the performance between weakly- and fully-supervised object detectors, and outperforming all methods mixing many weak and a few strong annotations in the low annotation regime.

In this work, we have combined weakly-supervised and active learning for reducing human annotation effort for object detectors. There are other types of methods that require no annotation at all, such as the unsupervised object discovery methods presented in the previous chapters (OSD, rOSD, LOD and LOST) and self-supervised pre-training [Caron, 2021; Chen, 2020c], and they might help improving different components of our pipeline, e.g., region proposals or the detection architecture. Future work will be dedicated to improving our approach by following those directions.

# Chapter 7

# Conclusions

## 7.1 Contributions

In this thesis, we have developed several annotation-efficient approaches to the localization and detection of objects in images.

We first considered the unsupervised object discovery problem which aims to localize objects in each image of an image collection and link images that contain similar objects without any annotation available. This is a challenging problem due to the lack of supervision and the ambiguity of object definition. We have discussed several approaches to this problem: OSD, rOSD, LOD and LOST.

In OSD, we cast unsupervised object discovery as selecting the best region from each image amongst a set of region proposals generated with off-the-shelf methods [Uijlings, 2013; Manen, 2013; Zitnick, 2014], and formulated it as a discrete optimization problem. We showed that this problem can be relaxed into a convex optimization problem and an approximate solution can be found by solving its dual problem with gradient descent. Alternatively, we can also find a solution with a greedy block-coordinate ascent procedure directly on the original formulation. This formulation was shown to be more robust to different types of region proposals and significantly outperforms the previous state of the art [Cho, 2015].

rOSD is built on OSD and addresses its limitations: The reliance on randomized Prim [Manen, 2013], a type of partially-supervised region proposals for good performance; the unsatisfactory performance with more powerful CNN features; the limited ability to discover multiple objects per image due to the presence of nearly-duplicated region proposals; and the high computational cost which limits its application on large datasets. To this end, we have made several contrbutions in rOSD. We introduced a novel algorithm for generating region proposals directly from pre-trained CNN features. This algorithm exploits a known observation on pre-trained CNN features for image classification and produces region proposals with a high rate of positive ones – those that highly overlap with ground-truth objects – which proves to be important for good performance in rOSD. These region proposals also have a nice intrinsic group property that we leveraged as additional regularizers to the original OSD formuation. These regularizers help to diversify regions returned by rOSD and enable effective multi-object discovery. Finally, we proposed an efficient two-stage algorithm that allows the applications of unsupervised object

discovery on datasets several times larger than those previously considered in OSD.

LOD does not follow the formulation of OSD and rOSD. Instead, it observes the analogy between our definition of objects in OSD and rOSD – objects are visual patterns that appear in multiple images – and the well-connected nodes in the graph of regions where nodes are region proposals and edges are weighted with region similarity. Finding the latter is thus equivalent to finding the former. This enables the application of existing ranking methods for finding well-connected nodes in graphs such as PageRank [Brin, 1998; Page, 1999] and eigenvector centrality [Landau, 1895]. These methods are highly efficient and parallelizable, permitting to scale unsupervised object discovery to datasets of millions of images. We also proposed a new ranking algorithm, that combines the eigenvector centrality and personalized PageRank. We showed that this algorithm outperforms the two individual ranking methods and yields state-of-the-art performance for unsupervised object discovery. Running LOD with self-supervised features [Gidaris, 2021], eliminating the need for supervised region proposals (OSD) or supervised pre-trained features for classification (rOSD), we also demonstrated a viable completely unsupervised pipeline for object discovery.

LOST neither defines objects as in OSD, rOSD and LOD nor leverages information from multiple images. Instead, it relies on the power of the Transformer-based self-supervised features DINO [Caron, 2021] which are shown to contain explicit object location information. To tackle unsupervised object discovery, we proposed in LOST to first use a simple seed-growing procedure to find for each image an object. We then used these objects as pseudo-labels for training a class-agnostic or a class-aware object detector. We showed that this simple method outperforms all the previous methods (OSD, rOSD and LOD). The class-agnostic detector is able to discover multiple objects per image while the class-aware detector can also group similar images together. It even competes with weakly-supervised object detectors which are trained with ground-truth image class labels.

The approaches for unsupervised object discovery in this thesis have pushed its limits gradually, from discovering only one object to finding multiple objects per image, from being applicable only on small datasets of thousands of images to being able to handle collections of millions of images, and from expensive optimization-based methods to a light-weight algorithm that leverages recent powerful self-supervised features.

It is important to push the limit of unsupervised methods but, in practice, we often have access to some form of supervision. We considered in BiB such a practical scenario for training object detectors where weak labels (image classes) are readily available and a small budget for full annotation is also available. Weakly-supervised object detection is an attractive alternative to its fully-supervised counterpart since it requires a much cheaper form of annotation, but weakly-supervised detectors still lag behind fully-supervised counterparts and suffer some known forms of confusion [Ren, 2020a]. We proposed to select several images that are then annotated with bounding boxes and fine-tune weakly-supervised detectors with the newly acquired annotation. The selection is done with BiB, an active learning strategy that chooses images on which the model is most confused, where model confusion is gauged through BiB pairs – pairs of model predictions, one of which is contained in the other. We showed that repeating this process

several times, we can improve significantly the weakly-supervised detector's performance while having only small additional annotation cost.

## 7.2 Future Work

Our work on unsupervised object discovery in this thesis mainly focuses on images. Leveraging additional information from other modalities could improve discovery performance. An example is motion cues in video, which helps to distinguish moving objects from static ones and background [Kwak, 2015]. Readily available textual information such as captions for images in social network is another useful additional information. It can be used to extract similarities between images, *e.g.* whether they contain similar objects, which is important in framework such as OSD or rOSD. Sound has been used to better localize objects in a weakly-supervised setting [Liu, 2021a]. It is also desirable to use this modality for unsupervised object discovery.

Unsupervised object discovery methods output object-centric regions in images, which can be used as noisy free annotation for other tasks. They have been used as pseudo-labels to train models for weakly-supervised object localization [Zhang, 2020b]. It would be interesting to consider similar applications for related tasks such as weakly- or semi-supervised object detection. For example, in weakly-supervised object detection, learning often proceeds with training a multiple instance learning module with ground-truth class labels, then generating pseudo-labels for refining several detector heads. These pseudo-labels are very unreliable at the beginning and coupling them with the output of unsupervised object discovery would improve the final detector's performance.

Self-supervised feature learning methods often train a feature extractor by comparing the features of different views of the same image. These views are often obtained with random cropping [Chen, 2020b; Caron, 2020; Gidaris, 2021]. This practice could lead to the irrelevant comparisons between semantically unrelated views (two different objects/scenes or backgroud *vs.* foreground). It is interesting to explore the use of object-centric patches produced by unsupervised object discovery methods as views of the image in this scenario [Mishra, 2021]. Object discovery and self-supervised feature learning can also be combined together into a single model where the object discovery component provides more fine-grained information for training the self-supervised feature learning component, which in turns improves object discovery. [Hénaff, 2022] has recently explored this direction but more effort needs to be invested.

Finally, in this thesis, we have shown that combining active and weakly-supervised learning is an interesting approach to annotation-efficient learning, especially since it does not require a substantial amount of full annotation to begin with. In the case that we have considered, we leverage weak annotation as a starting point but this can be generalized to the case where no annotation is required at the beginning. With the rising effectiveness of self-supervised features, it would be interesting to apply active learning on models that are trained in a completely unsupervised manner by leveraging available information from self-supervised features.

# List of Publications

## Conference papers

**Huy V. Vo**, Oriane Siméoni, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Jean Ponce. "*Active Learning Strategies for Weakly-Supervised Object Detection*". European Conference on Computer Vision, Tel Aviv, 2022.

**Huy V. Vo**, Elena Sizikova, Cordelia Schmid, Patrick Pérez, Jean Ponce. "*Large-Scale Unsupervised Object Discovery*". Advances in Neural Information Processing Systems, Virtual, 2021.

Oriane Siméoni, Gilles Puy, **Huy V. Vo**, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Renaud Marlet, Patrick Pérez, Jean Ponce. "*Localizing Objects with Self-Supervised Transformers and no Labels*". British Machine Vision Conference, Virtual, 2021.

**Huy V. Vo**, Patrick Pérez, Jean Ponce. "*Toward unsupervised, multi-object discovery in large-scale image collections*". European Conference on Computer Vision, Virtual, 2020.

**Huy V. Vo**, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, Jean Ponce. "*Unsupervised Image Matching and Object Discovery as Optimization*". IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, 2019.

# Appendix

## A Unsupervised Image Matching and Object Discovery as Optimization

**Maximization of supermodular cubic pseudo-Boolean functions.** An immediate corollary of [Billionnet, 1985, Lemma 1] is that a cubic pseudo-Boolean function with nonegative trinary coefficients and no binary terms is supermodular. For fixed $\lambda$ and $\mu$, this is obviously the case for the Lagrangian $K$ in Equation (2.5).

In addition, the unary terms in $K$ are nonpositive, and the Langragian can thus be rewritten, up to some constant additive term, in the form

$$f(x_1, \ldots, x_n) = \sum_{i \in U} c_i \bar{x}_i + \sum_{(i,j,k) \in T} c_{ijk} x_i x_j x_k, \tag{A.1}$$

where $\bar{x}_i = 1 - x_i$ (the *complement* of $x_i$), $U \subset \{1, \ldots, n\}$, $T \subset \{1, \ldots, n\}^3$, and all coefficients $c_i$ and $c_{ijk}$ are positive. We specialize in the rest of this section the general maximization method of [Billionnet, 1985] to functions of this form.

The *conflict graph* [Billionnet, 1985; Boros, 2002] $G(f)$ associated with such a function $f$ has as a set of nodes $X(f) = V \cup W$, where the elements of $V$ correspond to linear terms, those of $W$ correspond to cubic terms, and an edge links to nodes when one of the corresponding terms contains a variable, and the other one its complement. By construction $G(f)$ is a bipartite graph, with edges joining only elements of $V$ to elements of $W$.

As shown in [Billionnet, 1985] maximizing $f$ amounts to finding a maximum weight stable set in $G(f)$, where the nodes of $V$ are assigned weights $c_i$ and the nodes of $W$ are assigned weights $c_{ijk}$, which in turn reduces to computing a maximum flow between nodes $s$ and $t$ in the network deducted from $G(f)$ by (1) adding a source node and edges with upper capacity bound $c_i$ between $s$ and the corresponding elements of $V$; (2) adding a sink node $t$ and edges with upper capacity bound $c_{ijk}$ between the corresponding elements of $W$ and $t$; (3) assigning to all edges (from $V$ to $W$) in $G(f)$ an upper capacity bound of $+\infty$.

Let $[A, \bar{A}]$ denote the minimum cut obtained by computing the maximum flow in this graph, where $s$ is an element of $A$ and $t$ is an element of $\bar{A} = X(f) \setminus A$. The maximum weight stable set is then $S = (A \cap V) \cup (\bar{A} \cap W)$. The monomials $\bar{x}_i$ and $x_i x_j x_k$ associated with elements of $S$ are set to 1, from which the values of all variables are easily deduced.

# B Toward Unsupervised, Multi-Object Discovery in Large-Scale Image Collections

**Full results with both VGG16 and VGG19 features.** We present in Tables B.1, B.2 and B.3 our full results in colocalization and object discovery with features from both VGG16 and VGG19. It can be seen that, with VGG16 features, rOSD still significantly outperforms OSD on the two large datasets and fares comparably to OSD on the smaller two. It is also noticeable that rOSD significantly outperforms Wei *et al.* in both colocalization and single-object discovery on all datasets when VGG16 features are used.

| Method | Features | OD | VOc_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| [Cho, 2015] | WHO | 84.2 | 67.6 | 37.6 | - |
| OSD† | WHO | 87.1 ± 0.5 | 71.2 ± 0.6 | 39.5 ± 0.1 | - |
| [Li, 2016] | VGG16 | - | - | 40.0 | 41.9 |
| [Wei, 2019] | VGG16 | 86.9 | 66.2 | 44.7 | 47.6 |
| Ours (OSD) | VGG16 | 89.0 ± 0.6 | 73.6 ± 0.6 | 44.7 ± 0.3 | 49.0 ± 0.2 |
| Ours (rOSD) | VGG16 | 89.0 ± 0.5 | 73.3 ± 0.5 | 45.8 ± 0.3 | 49.7 ± 0.1 |
| [Li, 2016] | VGG19 | - | - | 41.9 | 45.6 |
| [Wei, 2019] | VGG19 | 87.9 | 67.7 | **48.7** | **51.1** |
| Ours (OSD) | VGG19 | **90.3 ± 0.3** | 75.3 ± 0.7 | 45.6 ± 0.3 | 47.8 ± 0.2 |
| Ours (rOSD) | VGG19 | 90.2 ± 0.3 | **76.1 ± 0.7** | 46.7 ± 0.2 | 49.2 ± 0.1 |

Table B.1 – Single-object colocalization performance of our approach compared to the state of the art. Note that Wei *et al.* [Wei, 2019] outperform our method on VOC_all and VOC12 with VGG19 features in this case, but the situation is clearly reversed in the much more difficult single-object discovery setting, as demonstrated in Table B.2. OSD† denotes the original OSD in Chapter 2.

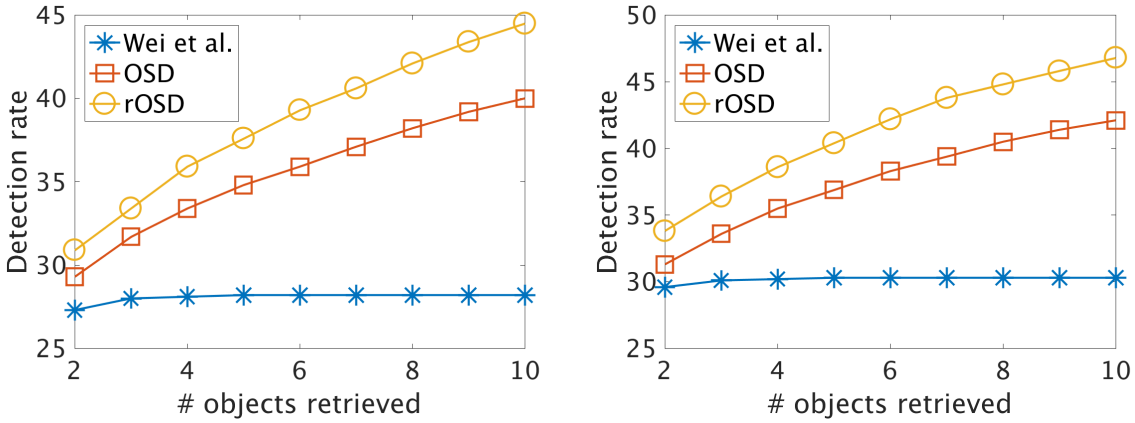
| Method | Features | OD | VOC_6x2 | VOC_all | VOC12 |
|---|---|---|---|---|---|
| [Cho, 2015] | WHO | 82.2 | 55.9 | 37.6 | - |
| OSD† | WHO | 82.3 ± 0.3 | 62.5 ± 0.6 | 40.7 ± 0.2 | - |
| [Wei, 2019] | VGG16 | 73.5 | 66.2 | 41.9 | 45.0 |
| Ours (OSD) | VGG16 | 87.8 ± 0.4 | 69.2 ± 0.5 | 48.7 ± 0.3 | 51.3 ± 0.2 |
| Ours (rOSD) | VGG16 | 87.6 ± 0.3 | 71.1 ± 0.8 | 49.2 ± 0.2 | **52.1 ± 0.1** |
| [Wei, 2019] | VGG19 | 75.0 | 54.0 | 43.4 | 46.3 |
| Ours (OSD) | VGG19 | 89.1 ± 0.4 | 71.9 ± 0.7 | 47.9 ± 0.3 | 49.2 ± 0.2 |
| Ours (rOSD) | VGG19 | **89.2 ± 0.4** | **72.5 ± 0.5** | **49.3 ± 0.2** | 51.2 ± 0.2 |

Table B.2 – Single-object discovery performance in the mixed setting on the datasets with our proposals compared to the state of the art. OSD† denotes the original OSD in Chapter 2.

**Multi-object experiments.** For a fair comparison to OSD and Wei *et al.* [Wei, 2019] in multi-object discovery, we have fixed the number of objects retained in each image by all methods to 5. We have also modified the method of Wei *et al.* such that 5 bounding boxes around the 5 largest clusters of positive pixels in their *indicator matrix* are returned as objects. For OSD

| Method | Features | Colocalization | | Discovery | |
|---|---|---|---|---|---|
| | | VOC_all | VOC12 | VOC_all | VOC12 |
| OSD† | WHO | 40.7 ± 0.1 | - | 30.7 ± 0.1 | - |
| [Wei, 2019] | VGG16 | 38.3 | 40.4 | 25.8 | 28.2 |
| Ours (OSD) | VGG16 | 45.9 ± 0.1 | 48.1 ± 0.0 | 34.9 ± 0.1 | 37.6 ± 0.0 |
| Ours (rOSD) | VGG16 | 48.5 ± 0.1 | 50.7 ± 0.1 | 37.2 ± 0.1 | **40.8 ± 0.1** |
| [Wei, 2019] | VGG19 | 43.3 | 45.5 | 28.1 | 30.3 |
| Ours (OSD) | VGG19 | 46.8 ± 0.1 | 47.9 ± 0.0 | 34.8 ± 0.0 | 36.9 ± 0.0 |
| Ours (rOSD) | VGG19 | **49.4 ± 0.1** | **51.5 ± 0.1** | **37.6 ± 0.1** | 40.4 ± 0.1 |

Table B.3 – Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets. OSD† denotes the original OSD in Chapter 2.




(a) VOC_all (b) VOC12

Figure B.1 – Multi-object discovery performance of rOSD compared to OSD and [Wei, 2019] when varying the maximum number of returned objects.

and rOSD, we run the corresponding optimization then apply the following post processing on each image: all $\nu$ retained regions are ranked in descending order using the score proposed in Equation (2.14), which is solely based on their similarity to the retained regions in the image's neighbors; We then iteratively discard all proposals having an IoU score greater than some threshold with higher-ranked regions; Among remaining regions, we return the 5 highest ranked as retrieved objects. Since this procedure can eliminate all but a few regions if the regions highly overlap, we choose a large value of $\nu$ (50) and a large value of $IoU$ threshold (0.7) in our experiments to guarantee that we have *exactly* 5 objects. This is, however, just a design choice and one can choose to retain fewer or more regions. We have conducted experiments with the number of retrieved objects varied in the interval $[2, 10]$ and observed that rOSD always yields better performance than OSD and [Wei, 2019] regardless of the number of objects retrieved (Figure B.1).

To eliminate overlapping boxes and obtain better qualitative results for OSD and rOSD, we have conducted experiments with $\nu = 25$ in the optimization and $IoU = 0.3$ for suppression threshold in the post processing. We have shown the qualitative improvements of this change

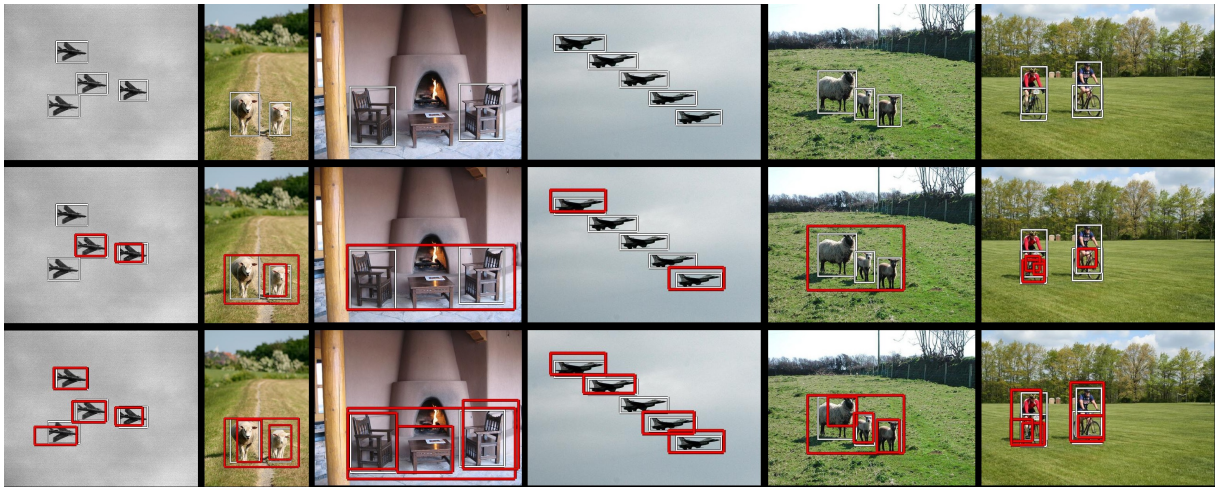


Figure B.2 – Multi-object discovery results. In each column, from top to bottom: original image, image with predictions of OSD, image with predictions of rOSD. White boxes are ground-truth objects and red ones are our predictions. There are *at most* 5 predictions per image.

| Method | Features | Colocalization | | Discovery | |
|---|---|---|---|---|---|
| | | VOC_all | VOC12 | VOC_all | VOC12 |
| [Wei, 2019] | VGG19 | 43.1 | 45.3 | 27.8 | 30.0 |
| Ours (OSD) | VGG19 | 39.6 ± 0.1 | 41.6 ± 0.1 | 29.0 ± 0.1 | 31.3 ± 0.1 |
| Ours (rOSD) | VGG19 | **47.3 ± 0.1** | **49.3 ± 0.1** | **36.7 ± 0.1** | **39.2 ± 0.1** |

Table B.4 – Multi-object colocalization and discovery performance of rOSD compared to competitors on VOC_all and VOC12 datasets when using smaller values of $\nu$ (25) and $IoU$ (0.3) threshold.

in Figure 3.7. We show here in Figure B.2 a qualitative comparison between the two methods. It can be seen that rOSD fares much better than OSD in localizing multiple objects. We also compare the quantitative performance of rOSD, OSD and [Wei, 2019] in this case in Table B.4. For [Wei, 2019], we take as before the bounding boxes around the largest clusters of pixels in the *indicator matrix* of each image. The number of clusters in this case is chosen to be the number of objects returned by rOSD in the same image. The results show that rOSD again yields by far the best performance. It is also noticeable that while using smaller values of $\nu$ and the $IoU$ threshold slightly deteriorates the performance of rOSD, it makes the performance of OSD drop significantly (compare Tables B.3 and B.4). This is due to the fact that OSD returns many highly overlapping regions and most of them are eliminated by our procedure. On the other hand, rOSD returns more diverse regions and consequently more regions are retained. In practice, we observe that OSD returns on average 1.47 (respectively 1.52) regions while rOSD returns 3.62 (respectively 3.63) on VOC_all (respectively VOC12). Note, however, that rOSD still outperforms OSD and [Wei, 2019] even when the latter are allowed to retain exactly 5 regions.

**Evaluating the graph computed by OSD.** Following [Cho, 2015], we evaluate the local graph structure obtained by rOSD using the CorRet measure, defined as the average percentage

of returned image neighbors that belong to the same (ground-truth) class as the image itself. As a baseline, we consider the local graph induced by the sets of nearest neighbors $N(i)$ computed from the fully connected layer *fc6* of the CNN that are used in the same experiment. Table B.5 shows the CorRet of local graphs obtained when running rOSD (OSD) on VOC_all and VOC12 and large-scale rOSD (OSD) on C20K in the mixed setting. It can be seen that the local image graphs returned by our methods have higher CorRet than the baseline.

| Dataset | VOC_all | VOC12 | C20K |
|---|---|---|---|
| Baseline | 50.7 | 56.4 | 36.8 |
| Ours (OSD) | **60.1 ± 0.1** | **63.2 ± 0.0** | **39.8 ± 0.0** |
| Ours (rOSD) | 59.8 ± 0.1 | 63.0 ± 0.0 | 39.4 ± 0.0 |

Table B.5 – Quality of the returned local image graph as measured by CorRet.

# C Large-Scale Unsupervised Object Discovery

**More qualitative results.** We show additional examples on COCO [Lin, 2014] in Figure C.1 and on OpenImages [Krasin, 2017] in Figure C.2 for which LOD successfully discovers objects. We also present some failure cases in Figure C.3. LOD typically fails to discover objects that are too small (images 1 to 5) or only discovers the most discriminative object parts instead of entire objects (images 6 to 8). In some cases, LOD discovers objects that are not annotated: entrance in image 1, tower in image 2 and flower branch in image 5.

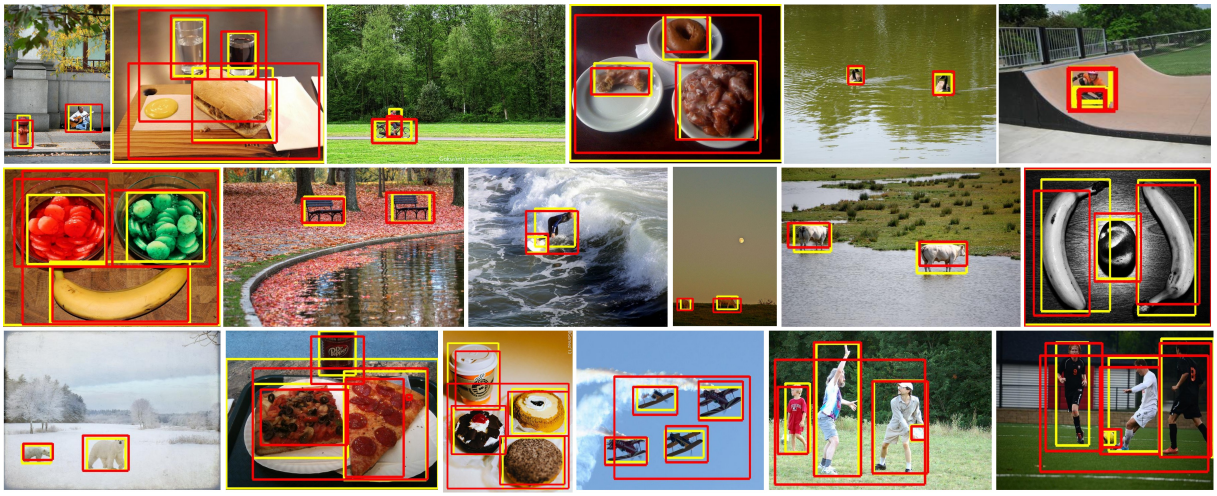


Figure C.1 – Examples in the COCO [Lin, 2014] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.



Figure C.2 – Examples in the OpenImages [Krasin, 2017] dataset where LOD successfully discovers ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.

**Multi-object discovery performance according to a detection rate metric.** Contrary to rOSD, we have evaluated the multi-object discovery performance of LOD using object discovery Average Precision (odAP) instead of detection rate (DetRate), which can also be thought of

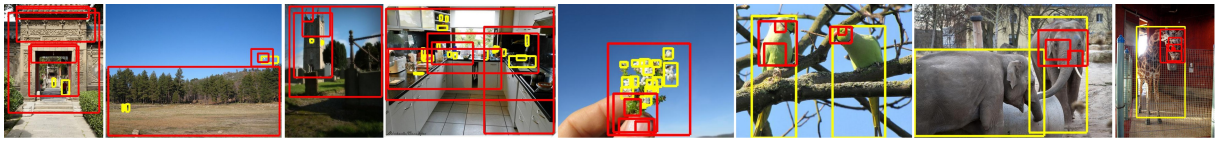


Figure C.3 – Examples in the COCO [Lin, 2014] and OpenImages [Krasin, 2017] datasets where LOD fails to discover ground-truth objects. Ground-truth boxes are in yellow and our predictions are in red.

recall over ground-truth objects. We argue in Chapter 4 that plain DetRate is not a good metric for multi-object discovery since it depends on the number $m$ of regions returned per image, which is pre-defined. Beside the fact that there is *a priori* no optimal choice for $m$, evaluating the performance at a single value of $m$ does not capture the range of possible performances. odAP, on the other hand, summarizes the performance at different values of $m$.

Despite these remarks, we present here for completeness the multi-object discovery performance in DetRate for LOD and the baselines in Table C.1. In addition to computing DetRate at $m = 5$ as in rOSD, we also consider $m = \bar{m}$ where $\bar{m}$ is the average number of ground-truth objects per image in the dataset, which is 7 for C20K and C120K, and 8 for Op50K and Op1.7M. The results show that LOD significantly outperforms the baselines in all datasets when detection rate is computed at $m = 5$. It also performs better than the others when detection rate is computed at $m = \bar{m}$, except for Edgeboxes [Zitnick, 2014] on Op1.7M dataset. However, we stress again that we think odAP is a more appropriate metric for multi-object discovery than DetRate. We show in Chapter 4 that LOD is significantly and consistently better than all baselines in all datasets according to odAP.

| Method | DetRate ($m = 5$) | | | | DetRate ($m = \bar{m}$) | | | |
|---|---|---|---|---|---|---|---|---|
| | C20K | C120K | Op50K | Op1.7M | C20K | C120K | Op50K | Op1.7M |
| [Zitnick, 2014] | 12.0 | 12.1 | 12.5 | 12.5 | 14.5 | 14.5 | 16.0 | **16.0** |
| [Wei, 2019] | 6.8 | 6.9 | 5.7 | 5.7 | 6.8 | 6.9 | 5.7 | 5.7 |
| [Kim, 2009] | 10.5 | 10.6 | 10.8 | - | 12.1 | 12.2 | 12.9 | - |
| rOSD | 12.3 | 11.8 | 11.8 | - | 13.3 | 12.7 | 13.1 | - |
| Ours (LOD) | **14.2** | **14.2** | **14.0** | **13.7** | **15.7** | **15.7** | **16.2** | 15.8 |

Table C.1 – Large-scale multi-object discovery performance and comparison to the state of the art on COCO [Lin, 2014], OpenImages [Krasin, 2017] and their respective subsets C20K and Op50K, as measured by detection rate.

**Proof of Lemma 1.**

*Proof.* Since $W$ is symmetric, all its eigenvalues are real and it can be diagonalized by an orthonormal basis of its eigenvectors. The maximizer of $t^T W t$ in the unit ball is the unit eigenvector of $W$ associated with its largest eigenvalue $\lambda^*$. Given that $W$ is irreducible, it has a unique, unit, non-negative eigenvector associated with its largest eigenvalue, according to the Perron-Frobenius theorem [Frobenius, 1912; Perron, 1907]. ■

**Note:** This is a classic result, only included here for completeness.

# D Localizing Objects with Self-Supervised Transformers and no Labels

**Analysis of DINO-seg.** In this section, we investigate alternative setups of the baseline DINO-seg which is based on [Caron, 2021]. They are presented in Table D.1. First, instead of using the best attention head over the entire dataset (as we did in Chapter 5), we evaluate the localization accuracy of DINO-seg for each one of the 6 available heads. We find out that one head in particular, namely head 4, captures objects well, whilst results with other heads are much lower. Due to its superior performance, in Chapter 5 we report DINO-seg using head 4.

We also explore dynamically selecting one box per image among boxes corresponding to the different heads using some heuristics. We report the two variants that gave the best results. In the first variant, we consider selecting the box corresponding to the head with the biggest connected component ('DINO-seg BCC'). However, it yields worse results than with head 4. We also try selecting, over the 6 boxes of the different heads, the box that has the highest average IoU overlap with the remaining 5 boxes ('DINO-seg HAIoU'). It improves over DINO-seg [head 4] by 1 point on both VOC07 and VOC12. However, as shown in Table D.1, it still performs significantly worse than LOST in this single-object discovery task.

| Method | VOC07_trainval | VOC12_trainval | C20K |
|---|---|---|---|
| DINO-seg [head 0] | 25.9 | 24.6 | 30.1 |
| DINO-seg [head 1] | 36.2 | 35.9 | 35.8 |
| DINO-seg [head 2] | 32.1 | 33.2 | 31.6 |
| DINO-seg [head 3] | 21.6 | 20.0 | 26.3 |
| DINO-seg [head 4] | 45.8 | 46.2 | **42.1** |
| DINO-seg [head 5] | 35.5 | 42.1 | 26.5 |
| DINO-seg BCC | 38.8 | 45.2 | 28.8 |
| DINO-seg HAIoU | **46.1** | **47.6** | 40.8 |
| LOST (ours) | **61.9** | **64.0** | **50.7** |

Table D.1 – DINO-seg ablation study. We compare here CorLoc results on datasets VOC07_trainval, VOC12_trainval and C20K when applying the DINO-seg method to create a box from the different heads of the attention layer. Also, DINO-seg BCC selects the box/head that produces the biggest connected component, and DINO-seg HAIoU selects the box/head that has the highest average IoU with the other 5 boxes. We additionally report results with our method LOST for comparison.

**Results on more datasets used in previous work.** For completeness, we present in Table D.2 results on the datasets used in [Wei, 2019], OSD and rOSD. In particular, we evaluate our method on the datasets VOC07_noh and VOC12_noh datasets (also named VOC_all and VOC12 in rOSD). They are subsets of the trainval set of the well-known PASCAL VOC 2007 and PASCAL VOC 2012 datasets containing 3550 and 7838 images respectively. These subsets exclude all images containing only objects annotated as "hard" or "truncated" and all boxes annotated as "hard" or "truncated".

| Method | VOC07_noh | VOC12_noh |
|---|---|---|
| OSD | 40.7 | - |
| [Wei, 2019] | 43.4 | 46.3 |
| rOSD | 49.3 | 51.2 |
| LOD | 48.0 | 50.5 |
| LOST | 54.9 | 57.5 |

Table D.2 – CorLoc results on the VOC07_noh and VOC12_noh datasets.

| Method | Features | CorLoc (%) | | |
|---|---|---|---|---|
| | | VOC07_trainval | VOC12_trainval | C20K |
| LOD | VGG16 | 53.6 | 55.1 | 48.5 |
| LOD | DINO | 43.2 | 45.9 | 33.7 |
| LOST | VGG16 | 42.0 | 47.2 | 30.2 |
| LOST | DINO | **61.9** | **64.0** | **50.7** |

Table D.3 – Single-object discovery performance in CorLoc of LOD and LOST with different types of features.

**Using DINO features for LOD.** We are aware that, in Table 5.1, we compare LOST using a transformer backbone to methods based on a VGG16 pre-trained on ImageNet models. For a fair comparison, we investigate here LOD when adapted to use the transformers features.

LOD uses the algorithm from rOSD to generate region proposals from CNN features, but we observe that this algorithm does not yield good proposals with transformer features. We therefore run LOD with edgeboxes [Zitnick, 2014] and use DINO [Caron, 2021] features, extracted with RoIPool [Girshick, 2015], to represent these proposals. We present the results on VOC07_trainval, VOC12_trainval and C20K dataset in Table D.3. Our results in Table 5.2 show that a direct adaption of LOST, designed by analysing the properties of transformers features, to CNN features yields worse performance. Conversely, as we see in Table D.3 here, adapting algorithms developed using properties of CNN features to transformer features is also not direct. Nevertheless, the number of design choices to adapt these algorithms to new types of features is vast and we do not exclude that some design choices might improve the results even further, e.g., by exploiting together CNN and transformer features.

**Using supervised pre-training.** We test LOST but this time using a transformer pre-trained under full supervision on ImageNet. We use the model provided by DeiT [Touvron, 2020]. With this model, LOST achieves a CorLoc of 16.9% which is significantly worse than the results obtained with the DINO self-supervised pre-trained model. We remark that a similar observation was made for DINO [Caron, 2021], where the segmentation performance obtained with the model trained under full supervision yields significantly worse results than when using DINO's model. It is unclear, however, if this difference of performance can be attributed to the properties of the self-supervision loss or to the more aggressive data augmentation used during DINO pre-training.

**Training details of the Faster R-CNN detection models.** In Chapter 5, we explore the application of LOST in unsupervised object detection by using its pseudo-boxes as ground truth for training Faster R-CNN detection models. For the implementation of the Faster R-CNN detector, we use the `R50-C4` model of Detectron2 [Wu, 2019] that relies on a ResNet-50 [He, 2016] backbone. In our experiments, this ResNet-50 backbone is pre-trained with DINO self-supervision. Then, to train the Faster R-CNN model on the considered dataset, we use the protocol and most hyper-parameters from [He, 2020].

In details, we train with mini-batches of size 16 across 8 GPUs using `SyncBatchNorm` to finetune BatchNorm parameters, as well as adding an extra BatchNorm layer for the RoI head after `conv5`, i.e., `Res5ROIHeadsExtraNorm` layer in Detectron2. During training, the learning rate is first warmed-up for 100 steps to 0.02 and then reduced by a factor of 10 after 18K and 22K training steps. We use in total 24K training steps for all the experiments, except when training class-agnostic detectors on the pseudo-boxes of the VOC07 trainval set, in which case we use 10K steps. For all experiments, during training, we freeze the first two convolutional blocks of ResNet-50, i.e., `conv1` and `conv2` in Detectron2.

# E Active Learning Strategies for Weakly-Supervised Object Detection

**Visualization of BiB pairs.** Our selection method relies on the discovery of *box-in-box* (BiB) patterns. We provide in Figure E.1 more visualization of BiB pairs on images of VOC07 and COCO.

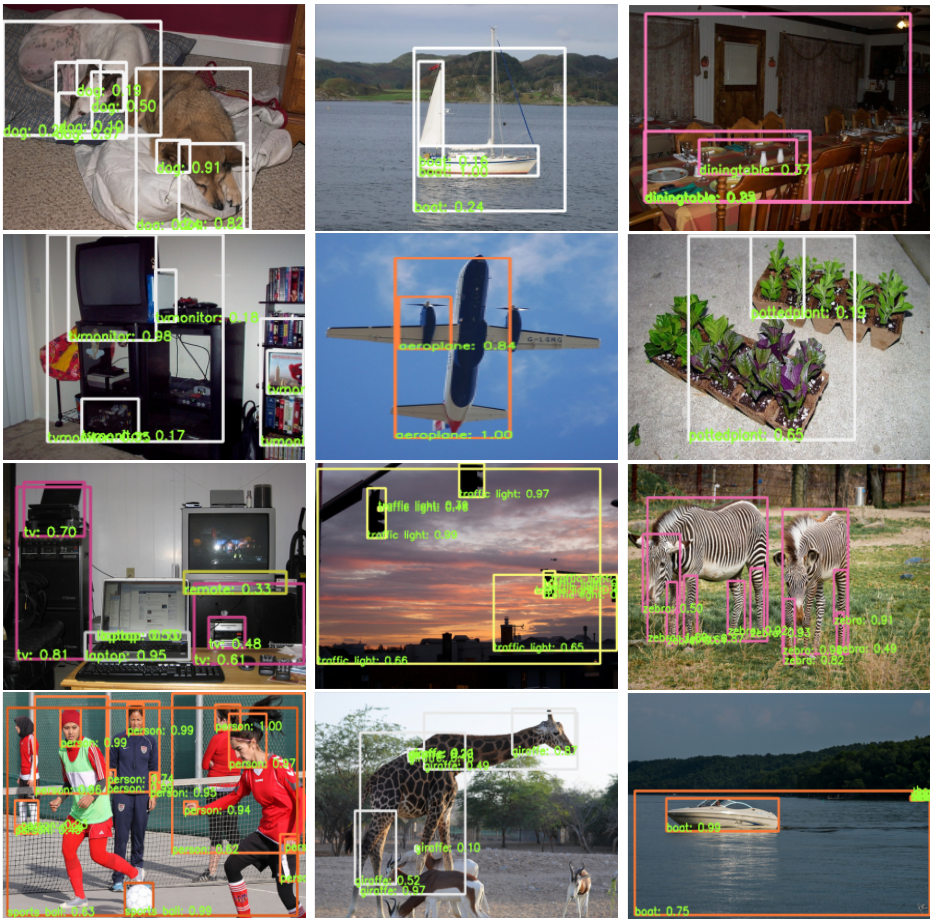


Figure E.1 – Examples of *box-in-box* (BiB) pairs on VOC07 (first two rows) and COCO (last two rows) extracted using the MIST [Ren, 2020a] detector.

**Detailed results of active learning strategies.** For experiments with active learning strategies, we have run each strategy six times on VOC07 and three times on COCO and reported the average performance in Chapter 6. For completeness, we provide in Tables E.1 and E.2 both the average and the standard deviation of the detector's performance in these experiments.

**Different variants of *loss*.** MIST [Ren, 2020a] is trained with a combination of losses coming from different heads. The Multiple Instance Learner produces $\mathcal{L}^{\text{MIL}}$ using the ground-truth class information while each refinement head $k \in \{1, 2, 3\}$ produces the refinement loss $\mathcal{L}_w^{(k)}$ using pseudo objects generated from the previous head. We have tested each of these losses and the combination of the three refinement losses $\sum_{k=1}^{3} \mathcal{L}_w^{(k)}$ in our experiments with *loss* strategy. We

| Method | Number of fully-annotated images | | | | |
|---|---|---|---|---|---|
| | 50 | 100 | 150 | 200 | 250 |
| u-random | 56.5 ± 0.4 | 58.4 ± 0.4 | 59.3 ± 0.7 | 60.2 ± 0.4 | 61.1 ± 0.5 |
| b-random | 56.7 ± 0.7 | 58.4 ± 0.7 | 59.7 ± 0.8 | 60.4 ± 0.5 | 61.2 ± 0.4 |
| core-set | 55.5 ± 0.6 | 57.7 ± 0.6 | 58.7 ± 0.5 | 59.5 ± 0.4 | 60.1 ± 0.2 |
| core-set-ent | 55.5 ± 0.4 | 57.6 ± 0.4 | 59.0 ± 0.4 | 60.0 ± 0.2 | 60.5 ± 0.2 |
| entropy-max | 57.0 ± 0.4 | 58.7 ± 0.2 | 59.6 ± 0.4 | 60.6 ± 0.2 | 60.9 ± 0.2 |
| entropy-sum | 56.5 ± 1.0 | 58.6 ± 0.4 | 59.8 ± 0.3 | 60.5 ± 0.5 | 61.2 ± 0.8 |
| loss | **59.7 ± 0.2** | 60.5 ± 0.5 | 61.3 ± 0.7 | 62.0 ± 0.5 | 62.5 ± 0.3 |
| BiB | 58.5 ± 0.8 | **60.8 ± 0.5** | **61.9 ± 0.4** | **62.9 ± 0.5** | **63.5 ± 0.4** |

Table E.1 – Comparison of active learning strategies on VOC07. For each experiment, we conducted 5 cycles with a budget of 50 images per cycle. We repeated the experiment six times for each strategy and report the average and standard deviation of their performance (in AP50). BiB yields significantly better performance than the others. *loss* performs well in the first cycle but fares worse than BiB in subsequent cycles. Additionally, it performs much worse, even than random, on COCO (see Table E.2).

| Method | AP | | | | | AP50 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 160 | 320 | 480 | 640 | 800 | 160 | 320 | 480 | 640 | 800 |
| u-random | 14.1 ± 0.1 | 15.1 ± 0.2 | 15.7 ± 0.2 | 16.1 ± 0.4 | 16.5 ± 0.3 | 29.1 ± 0.4 | 30.8 ± 0.3 | 31.7 ± 0.4 | 32.4 ± 0.4 | 33.0 ± 0.3 |
| b-random | 14.4 ± 0.4 | 15.2 ± 0.3 | 15.9 ± 0.1 | 16.2 ± 0.2 | 16.8 ± 0.2 | 29.5 ± 0.6 | 30.8 ± 0.4 | 31.8 ± 0.2 | 32.3 ± 0.1 | 33.3 ± 0.2 |
| entropy-sum | 12.3 ± 0.3 | 12.8 ± 0.2 | 13.3 ± 0.3 | 13.6 ± 0.4 | 13.7 ± 0.3 | 25.6 ± 0.4 | 26.5 ± 0.1 | 27.2 ± 0.2 | 27.7 ± 0.5 | 27.8 ± 0.1 |
| entropy-max | 12.7 ± 0.2 | 13.9 ± 0.1 | 14.5 ± 0.5 | 14.9 ± 0.3 | 15.2 ± 0.2 | 26.9 ± 0.2 | 28.9 ± 0.1 | 29.7 ± 0.5 | 30.4 ± 0.3 | 30.8 ± 0.3 |
| loss | 13.5 ± 0.1 | 14.1 ± 0.2 | 14.5 ± 0.2 | 14.7 ± 0.3 | 14.9 ± 0.3 | 27.8 ± 0.1 | 29.1 ± 0.1 | 29.7 ± 0.1 | 30.1 ± 0.3 | 30.4 ± 0.3 |
| core-set | 12.9 ± 0.2 | 14.5 ± 0.3 | 15.3 ± 0.2 | 15.9 ± 0.1 | 16.4 ± 0.3 | 26.9 ± 0.3 | 29.6 ± 0.5 | 30.9 ± 0.2 | 31.7 ± 0.2 | 32.5 ± 0.4 |
| core-set-ent | 13.1 ± 0.0 | 14.2 ± 0.1 | 15.1 ± 0.2 | 15.5 ± 0.3 | 16.0 ± 0.2 | 27.3 ± 0.2 | 29.2 ± 0.1 | 30.7 ± 0.2 | 31.3 ± 0.4 | 32.1 ± 0.2 |
| BiB | **14.8 ± 0.3** | **15.9 ± 0.2** | **16.5 ± 0.1** | **16.9 ± 0.2** | **17.2 ± 0.2** | **30.6 ± 0.1** | **32.4 ± 0.3** | **33.1 ± 0.2** | **33.8 ± 0.1** | **34.1 ± 0.1** |

Table E.2 – Comparison of active learning strategies on COCO. For each experiment, we conducted 5 cycles with a budget of 160 images per cycle. We repeated the experiment three times for each strategy and report the average and standard deviation of their performance (in AP50 and AP). BiB significantly outperforms all other methods.

present a summary of the results in Table E.3. For each experiment, we have conducted 5 cycles with a budget of 50 images per cycle on VOC07. On average, $\mathcal{L}_w^{(3)}$ yields the best results on this dataset and we use it for all experiments with the *loss* strategy in Chapter 6.

**Ablation study on COCO.** We have provided an ablation study on different components of BiB on VOC07 dataset in Chapter 6. For completeness, we report in Table E.4 the averaged AP50 scores (over 3 repetitions) of the ablation study on COCO. The results are similar to those obtained on VOC07, except for the difficulty-aware sampling, which helps with the u-random strategy but not always with BiB.

**MIST architecture.** We use MIST [Ren, 2020a] as our base weakly-supervised object detector. MIST follows OICR [Tang, 2017] and consists of a Multiple Instance Learner (MIL) trained to produce coarse detections which are then refined with several refinement heads using automatically-generated pseudo-boxes. We have given details about the refinement heads in Chapter 6 and provide here a description of the MIL head as well as the procedure to gen-

| AL method | Number of fully-annotated images 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| $\mathcal{L}^{\text{MIL}}$ | 57.1 ± 0.3 | 57.9 ± 0.2 | 58.4 ± 0.5 | 59.4 ± 0.2 | 60.0 ± 0.3 |
| $\mathcal{L}_w^{(1)}$ | 58.2 ± 0.4 | 58.5 ± 0.4 | 59.6 ± 0.7 | 60.3 ± 0.8 | 61.1 ± 0.5 |
| $\mathcal{L}_w^{(2)}$ | 59.4 ± 0.3 | **60.7 ± 0.2** | **61.4 ± 0.3** | 61.8 ± 0.3 | 62.4 ± 0.1 |
| $\mathcal{L}_w^{(3)}$ | 59.7 ± 0.2 | 60.5 ± 0.5 | 61.3 ± 0.7 | **62.0 ± 0.5** | **62.5 ± 0.3** |
| $\sum_{k=1,2,3} \mathcal{L}_w^{(k)}$ | **59.9 ± 0.4** | 60.6 ± 0.5 | 60.9 ± 0.5 | 61.6 ± 0.3 | 62.2 ± 0.6 |

Table E.3 – Performance of the loss strategy with different choices of the detector's loss on VOC07. For each experiment, we perform 5 cycles with a budget of 50 images per cycle. We have repeated the experiment six times for each strategy and report the average and standard deviation of their performance.

| DifS | K selection | | | AP50 | | | | |
|---|---|---|---|---|---|---|---|---|
| | im. | reg. | BiB | 160 | 320 | 480 | 640 | 800 |
| | | | | 29.0 | 30.6 | 31.4 | 32.3 | 32.8 |
| ✓ | | | | 29.1 | 30.8 | 31.7 | 32.4 | 33.0 |
| ✓ | ✓ | | | 29.2 | 30.7 | 31.6 | 32.3 | 32.9 |
| ✓ | | ✓ | | 30.5 | 31.6 | 32.6 | 33.5 | 34.1 |
| | | | ✓ | 30.7 | 32.3 | 33.2 | 33.7 | 34.2 |
| ✓ | | | ✓ | 30.6 | 32.4 | 33.1 | 33.8 | 34.1 |

Table E.4 – Ablation study on COCO. We show the average and standard deviation results over several runs in AP50 on COCO with 5 cycles and a budget $B = 160$. *DifS* stands for the difficulty-aware region sampling module. Images are selected by applying k-means++ init. (*K selection*) on image-level features (*im.*), confident predictions' features (*reg.*) or BiB pairs.

erate the pseudo-boxes. We consider an image $\mathbf{I}$, its class labels $\mathbf{q} \in \{0,1\}^C$ and the set of pre-computed region proposals $\mathcal{R} = \{r_1, r_2, \ldots, r_R\}$. Please note that we drop here the image index in order to ease understanding.

*Multiple instance learner.* MIL receives $\mathbf{I}$ and $\mathcal{R}$ as input and yields a class probability vector $\phi \in \mathbb{R}^C$. It is trained to classify the image with the Binary Cross Entropy (BCE) loss $\mathcal{L}_{\text{MIL}}$ on $\phi$:

$$\mathcal{L}_{\text{MIL}} = -\frac{1}{C}\sum_{c=1}^{C} q(c)\log(\phi(c)) + (1-q(c))\log(1-\phi(c)). \tag{E.1}$$

In MIST, class probabilities $\phi$ are obtained by aggregating scores in a region score matrix $\mathbf{s} \in \mathbb{R}^{R\times C}$ with $c \in \{1, .., C\}$:

$$\phi(c) = \sum_{i=1}^{R} \mathbf{s}(i,c), \tag{E.2}$$

where $\mathbf{s} = \mathbf{s}_c \odot \mathbf{s}_d$ is the point-wise product of a classification score matrix $\mathbf{s}_c \in \mathbb{R}^{R\times C}$ and a detection score matrix $\mathbf{s}_d \in \mathbb{R}^{R\times C}$. Matrices $\mathbf{s}_c$ and $\mathbf{s}_d$ are built by concatenating projected regions features extracted with the backbone network for each of the regions in $\mathcal{R}$. Matrix $\mathbf{s}_c$ is normalized row-wise with the softmax operation and models the class probabilities of the region proposals. Matrix $\mathbf{s}_d$, which is normalized column-wise, represents the relative objectness of the proposals with respect to the corresponding classes. Given those interpretations, $\mathbf{s}(i,c)$

expresses the likelihood that region $i$ is an object of class $c$.

*Pseudo-boxes generation.* MIST [Ren, 2020a] introduces a heuristic to generate the pseudo-boxes $\mathbf{D}^{(k-1)}$ that are used to train the refinement heads $k$. Such boxes are generated either from the region score matrix $\mathbf{s}$ of the MIL (giving $\mathbf{D}^{(0)}$) or the region classification score matrices $\mathbf{s}^{(k)}$ ($k = 1, 2, 3$) of the refinement heads (giving $\mathbf{D}^{(k)}$). In particular, for each ground-truth class $c$ in image $\mathbf{I}$, the corresponding column scores $[\mathbf{s}(1, c), \ldots, \mathbf{s}(R, c)]$ in $\mathbf{s}$ (or $\mathbf{s}^{(k)}$) are sorted in descending order. Then, given the top-15% region proposals with the highest scores, we select all boxes that do not have an IoU$\geq$ 0.3 with a higher-ranked region. Selected boxes for all classes are aggregated to construct the final set of pseudo-boxes.

# References


[Afouras, 2021] T. Afouras, Y. M. Asano, F. Fagan, A. Vedaldi et F. Metze, « Self-supervised object detection from audio-visual correspondence », *ArXiv*, 2021.

[Afouras, 2020] T. Afouras, A. Owens, J. S. Chung et A. Zisserman, « Self-supervised learning of audio-visual objects from video », *ECCV*, 2020.

[Agarwal, 2020] S. Agarwal, H. Arora, S. Anand et C. Arora, « Contextual diversity for active learning », *European Conference on Computer Vision*, Springer, 2020, p. 137–153.

[Ahn, 2019] J. Ahn, S. Cho et S. Kwak, « Weakly supervised learning of instance segmentation with inter-pixel relations », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Ahn, 2018] J. Ahn et S. Kwak, « Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Alexe, 2012] B. Alexe, T. Deselaers et V. Ferrari, « Measuring the objectness of image windows », *TPAMI*, t. 34, 2012.

[Amir, 2021] S. Amir, Y. Gandelsman, S. Bagon et T. Dekel, « Deep vit features as dense visual descriptors », *ArXiv preprint arXiv :2112.05814*, 2021.

[Arthur, 2007] D. Arthur et S. Vassilvitskii, « K-means++ : the advantages of careful seeding », *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, sér. SODA '07, New Orleans, Louisiana : Society for Industrial et Applied Mathematics, 2007, p. 1027–1035.

[Arun, 2019] A. Arun, C. Jawahar et M. P. Kumar, « Dissimilarity coefficient based weakly supervised object detection », *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Asano, 2019] Y. M. Asano, C. Rupprecht et A. Vedaldi, « Self-labelling via simultaneous clustering and representation learning », *ArXiv*, 2019.

[Ash, 2020] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford et A. Agarwal, « Deep batch active learning by diverse, uncertain gradient lower bounds », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[Babenko, 2014] A. Babenko, A. Slesarev, A. Chigorin et V. Lempitsky, « Neural codes for image retrieval », *ECCV*, 2014.

[Babenko, 2015] A. Babenko et V. Lempitsky, « Aggregating deep convolutional features for image retrieval », *ICCV*, 2015.

[Bach, 2013] F. Bach, « Learning with submodular functions : a convex optimization perspective », *Foundations and Trends in Machine Learning*, 2013.

[Bachman, 2019] P. Bachman, R. D. Hjelm et W. Buchwalter, « Learning representations by maximizing mutual information across views », *Advances in Neural Information Processing Systems*, 2019.

[Baek, 2020] K. Baek, M. Lee et H. Shim, « Psynet : self-supervised approach to object localization using point symmetric transformation », *AAAI*, 2020.

[Bai, 2012] Y. Bai et M. Tang, « Robust tracking via weakly supervised ranking SVM », *CVPR*, 2012.

[Ballard, 1981] D. Ballard, « Generalizing the Hough transform to detect arbitrary shapes », *Pattern Recognition*, 1981.

[Bardes, 2022] A. Bardes, J. Ponce et Y. LeCun, « Vicreg : variance-invariance-covariance regularization for self-supervised learning », *International Conference on Learning Representations (ICLR)*, 2022.

[Bautista, 2016] M. A. Bautista, A. Sanakoyeu, E. Sutter et B. Ommer, « Cliquecnn : deep unsupervised exemplar learning », *ArXiv*, 2016.

[Belkin, 2004] M. Belkin, I. Matveeva et P. Niyogi, « Regularization and semi-supervised learning on large graphs », *International Conference on Computational Learning Theory (COLT)*, 2004.

[Bell, 2016] S. Bell, L. Zitnick, K. Bala et R. Girshick, « Inside-outside net : detecting objects in context with skip pooling and recurrent neural networks », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Beluch, 2018] W. H. Beluch, T. Genewein, A. Nürnberger et J. M. Köhler, « The power of ensembles for active learning in image classification », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Bergamo, 2016] A. Bergamo, L. Bazzani, D. Anguelov et L. Torresani, « Self-taught object localization with deep networks », *WACV*, 2016.

[Bergmann, 2019] P. Bergmann, T. Meinhardt et L. Leal-Taixe, « Tracking without bells and whistles », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Berthelot, 2020] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang et C. Raffel, « Remixmatch : semi-supervised learning with distribution matching and augmentation anchoring », *Proceedings of the International Conference in Learning Rrepresentations (ICLR)*, 2020.

[Berthelot, 2019] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver et C. Raffel, « Mixmatch : a holistic approach to semi-supervised learning », *Advances in Neural Information Processing Systems (NeurIPS)*, t. 32, 2019.

[Bhojanapalli, 2021] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner et A. Veit, « Understanding robustness of transformers for image classification », *ArXiv*, 2021.

[Biffi, 2020] C. Biffi, S. G. McDonagh, P. H. S. Torr, A. Leonardis et S. Parisot, « Many-shot from low-shot : learning to annotate using mixed supervision for object detection », *ECCV*, 2020.

[Bilen, 2016] H. Bilen et A. Vedaldi, « Weakly supervised deep detection networks », *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Billionnet, 1985] A. Billionnet et M. Minoux, « Maximizing a supermodular pseudoboolean function : a polynomial algorithm for supermodular cubic functions », *Discrete Applied Mathematics*, t. 12, p. 1–11, 1985.

[Blake, 1987] A. Blake et A. Zisserman, « Visual reconstruction », *MIT Press*, 1987.

[Blum, 1973] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest et R. E. Tarjan, « Time bounds for selection », *Journal of Computer and System Sciences*, t. 7, n° 4, p. 448–461, 1973.

[Bojanowski, 2017] P. Bojanowski et A. Joulin, « Unsupervised learning by predicting noise », *International Conference on Machine Learning (ICML)*, 2017.

[Bojanowski, 2015] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce et C. Schmid, « Weakly-supervised alignment of video with text », *ICCV*, 2015.

[Boros, 2002] E. Boros et P. Hammer, « Pseudo-Boolean optimization », *Discrete Applied Mathematics*, t. 123, n° 1-3, p. 155–225, 2002.

[Boser, 1992] B. E. Boser, I. M. Guyon et V. N. Vapnik, « A training algorithm for optimal margin classifiers », *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 1992, p. 144–152.

[Brin, 1998] S. Brin et L. Page, « The anatomy of a large-scale hypertextual web search », *Computer Networks*, 1998.

[Brust, 2019] C.-A. Brust, C. Kading et J. Denzler, « Active learning for deep object detection », *VISAPP*, 2019.

[Bryson, 1961] A. E. Bryson, « A gradient method for optimizing multi-stage allocation processes », *Proceedings of the Harvard Univ. Symposium on digital computers and their applications*, 1961.

[Buhet, 2020] T. Buhet, E. Wirbel, A. Bursuc et X. Perrotton, « Plop : probabilistic polynomial objects trajectory planning for autonomous driving », *Conference on Robot Learning (CoRL)*, 2020.

[Burgess, 2019] C. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick et A. Lerchner, « Monet : unsupervised scene decomposition and representation », *ArXiv*, 2019.

[Cakir, 2019] F. Cakir, K. He, X. Xia, B. Kulis et S. Sclaroff, « Deep metric learning to rank », *CVPR*, 2019.

References


[Cao, 2017] Z. Cao, T. Simon, S.-E. Wei et Y. Sheikh, « Realtime multi-person 2d pose estimation using part affinity fields », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Carion, 2020] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov et S. Zagoruyko, « End-to-end object detection with transformers », *ECCV*, 2020.

[Caron, 2018] M. Caron, P. Bojanowski, A. Joulin et M. Douze, « Deep clustering for unsupervised learning of visual features », *ECCV*, 2018.

[Caron, 2019] M. Caron, P. Bojanowski, J. Mairal et A. Joulin, « Unsupervised pre-training of image features on non-curated data », *ICCV*, 2019.

[Caron, 2020] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski et A. Joulin, « Unsupervised learning of visual features by contrasting cluster assignments », 2020.

[Caron, 2021] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski et A. Joulin, « Emerging properties in self-supervised vision transformers », *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[Chattopadhay, 2018] A. Chattopadhay, A. Sarkar, P. Howlader et V. Balasubramanian, « Grad-cam++ : generalized gradient-based visual explanations for deep convolutional networks », *WACV*, 2018.

[Chazal, 2013] F. Chazal, L. J. Guibas, S. Y. Oudot et P. Skraba, « Persistence-based clustering in riemannian manifolds », *Journal of the ACM*, t. 60, n° 6, 41 :1–41 :38, 2013.

[Chen, 2019a] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy et D. Lin, « Hybrid task cascade for instance segmentation », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Chen, 2019b] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy et D. Lin, « MMDetection : open mmlab detection toolbox and benchmark », *ArXiv preprint arXiv :1906.07155*, 2019.

[Chen, 2018a] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy et A. L. Yuille, « Deeplab : semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs », *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, t. 40, n° 4, p. 834–848, 2018.

[Chen, 2018b] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff et H. Adam, « Encoder-decoder with atrous separable convolution for semantic image segmentation », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[Chen, 2021] L. Chen, T. Yang, X. Zhang, W. Zhang et J. Sun, « Points as queries : weakly semi-supervised object detection by points », 2021, p. 8819–8828.

[Chen, 2020a] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan et I. Sutskever, « Generative pretraining from pixels », *ICML*, 2020.

[Chen, 2020b] T. Chen, S. Kornblith, M. Norouzi et G. Hinton, « A simple framework for contrastive learning of visual representations », *ICML*, 2020.

[Chen, 2020c] X. Chen, H. Fan, R. Girshick et K. He, « Improved baselines with momentum contrastive learning », *ArXiv preprint arXiv :2003.04297*, 2020.

[Chen, 2014] X. Chen, A. Shrivastava et A. Gupta, « Enriching visual knowledge bases via object discovery and segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, p. 2035–2042.

[Cheng, 2022] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov et R. Girdhar, « Masked-attention mask transformer for universal image segmentation », 2022.

[Chitta, 2019] K. Chitta, J. M. Alvarez et A. Lesnikowski, « Large-scale visual active learning with deep probabilistic ensembles », *ArXiv preprint arXiv :1811.03575*, 2019.

[Cho, 2015] M. Cho, S. Kwak, C. Schmid et J. Ponce, « Unsupervised object discovery and localization in the wild : part-based matching with bottom-up region proposals », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[Choe, 2020] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata et H. Shim, « Evaluating weakly supervised object localization methods right », *CVPR*, 2020.

[Choe, 2019] J. Choe et H. Shim, « Attention-based dropout layer for weakly supervised object localization », *CVPR*, 2019.

[Choi, 2021] J. Choi, I. Elezi, H.-J. Lee, C. Farabet et J. M. Alvarez, « Active learning for deep object detection via probabilistic modeling », *ICCV*, 2021.

[Cinbis, 2017] R. Cinbis, J. Verbeek et C. Schmid, « Weakly supervised object localization with multi-fold multiple instance learning », *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[Clowes, 1971] M. B. Clowes, « On seeing things », *Artificial Intelligence*, p. 79–116, 1971.

[Cordonnier, 2020] J.-B. Cordonnier, A. Loukas et M. Jaggi, « On the relationship between self-attention and convolutional layers », *ICLR*, 2020.

[Cordts, 2016] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke et B. Roth Stefan and Schiele, « The cityscapes dataset for semantic urban scene understanding », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Cortes, 1995] C. Cortes et V. Vapnik, « Support-vector networks », *Machine Learning*, p. 273–297, 1995.

[Dalal, 2005] N. Dalal et B. Triggs, « Histogram of oriented gradients for human detection », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[Dean, 2012] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. Le et A. Ng, « Large scale distributed deep networks », *Advances in Neural Information Processing Systems (NeuRIPS)*, F. Pereira, C. Burges, L. Bottou et K. Weinberger, éds., t. 25, 2012.

[Deng, 2009] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li et L. Fei-Fei, « Imagenet : a large-scale hierarchical image database », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[Deo, 2018] N. Deo et M. M. Trivedi, « Convolutional social pooling for vehicle trajectory prediction », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, p. 1549–15 498.

[Desai, 2019] S. V. Desai, A. C. Lagandula, W. Guo, S. Ninomiya et V. N. Balasubramanian, « An adaptive supervision framework for active learning in object detection », *BMVC*, 2019.

[Deselaers, 2010] T. Deselaers, B. Alexe et V. Ferrari, « Localizing objects while learning their appearance », *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, 2010, p. 452–466.

[Diba, 2017] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash et L. Van Gool, « Weakly supervised cascaded convolutional networks », *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Dickmanns, 1988] E. D. Dickmanns et V. Graefe, « Dynamic monocular machine vision », *Machine Vision and Applications (MVA)*, p. 223–240, 1988.

[Dietterich, 1997] T. G. Dietterich, R. H. Lathrop et T. Lozano-Pérez, « Solving the multiple instance problem with axis-parallel rectangles », *Artificial Intelligence*, t. 89, n° 1–2, p. 31–71, 1997.

[Doersch, 2015] C. Doersch, A. Gupta et A. Efros, « Unsupervised visual representation learning by context prediction », *ICCV*, 2015, p. 1422–1430.

[Dosovitskiy, 2021] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit et N. Houlsby, « An image is worth 16x16 words : transformers for image recognition at scale », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[Dreyfus, 1962] S. Dreyfus, « The numerical solution of variational problems », *Journal of Mathematical Analysis and Applications*, 1962.

[Ducoffe, 2018] M. Ducoffe et F. Precioso, *Adversarial active learning for deep networks : a margin based approach*, 2018.

[Ebrahimi, 2020] S. Ebrahimi, W. Gan, K. Salahi et T. Darrell, « Minimax active learning », *ArXiv*, t. abs/2012.10467, 2020.

[Ebrahimi, 2019] S. Ebrahimi, S. Sinha et T. Darrell, « Variational adversarial active learning », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Edelsbrunner, 2009] H. Edelsbrunner et J. Harer, « Computational topology : an introduction », *AMS Press*, 2009.

[Edelsbrunner, 2002] H. Edelsbrunner, D. Letscher et A. Zomorodian, « Topological persistence and simplification. », *Discrete and Computational Geometry*, 2002.

[Elezi, 2021] I. Elezi, Z. Yu, A. Anandkumar, L. Leal-Taixe et J. M. Alvarez, « Not all labels are equal : rationalizing the labeling costs for training object detection », *ArXiv*, t. abs/2106.11921, 2021.

[ElNouby, 2021] A. El-Nouby, N. Neverova, I. Laptev et H. Jegou, « Training vision transformers for image retrieval », *ArXiv*, 2021.

[Engelcke, 2020] M. Engelcke, A. Kosiorek, O. P. Jones et I. Posner, « Genesis : generative scene inference and sampling with object-centric latent representations », *ICLR*, 2020.

[Everingham, 2012] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn et A. Zisserman, *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, 2012.

[Everingham, 2007] M. Everingham, L. Van Gool, C. K. Williams, J. Winn et A. Zisserman, *The pascal visual object classes challenge 2007 (voc2007) results*, 2007.

[Faktor, 2012] A. Faktor et M. Irani, « Clustering by composition–unsupervised discovery of image categories », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[Fan, 2020] Q. Fan, W. Zhuo, C.-K. Tang et Y.-W. Tai, « Few-shot object detection with attention-rpn and multi-relation detector », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, p. 4013–4022.

[Fang, 2019] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li et C. Lu, « Instaboost : boosting instance segmentation via probability map guided copy-pasting », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, p. 682–691.

[Fang, 2020] L. Fang, H. Xu, Z. Liu, S. Parisot et Z. Li, « Ehsod : cam-guided end-to-end hybrid-supervised object detection with cascade refinement », *Proceedings of the AAAI Conference on Artificial Intelligence*, t. 34, n° 07, p. 10 778–10 785, 2020.

[FeiFei, 2006] L. Fei-Fei, R. Fergus et P. Peron, « One-shot learning of object categories », 2006.

[Felzenszwalb, 2010] P. Felzenszwalb, R. Girshick, D. McAllester et D. Ramanan, « Object detection with discriminatively trained part-based models », *IEEE Trans. Pattern Anal. and Machine Intell.*, t. 32, n° 9, p. 1627–1645, 2010.

[Feng, 2011] J. Feng, Y. Wei, L. Tao, C. Zhang et J. Sun, « Salient object detection by composition », *ICCV*, 2011.

[Fischler, 1973] M. A. Fischler et R. A. Elschlager, « The representation and matching of pictorial structures », *IEEE Transactions on Computers*, t. 22, n° 1, p. 67–92, 1973.

[Foulds, 2010] J. R. Foulds et E. Frank, « A review of multi-instance learning assumptions », *The Knowledge Engineering Review*, t. 25, p. 1–25, 2010.

[Frank, 1956] M. Frank et P. Wolfe, « An algorithm for quadratic programming », *Naval Research Logistics Quarterly*, t. 3, p. 95–110, 1956.

[Frobenius, 1912] F. G. Frobenius, « Über matrizen aus nicht negativen elementen », 1912.

[Fukushima, 1980] K. Fukushima, « Neocognitron : a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position », *Biol. Cybernetics*, t. 36, p. 193–202, 1980.

[Funkhouser, 2003] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin et D. Jacobs, « A search engine for 3D models », *ACM Trans. on Graphics*, 2003.

[Gal, 2017] Y. Gal, R. Islam et Z. Ghahramani, « Deep bayesian active learning with image data », *Proceedings of the International Conference on Machine Learning (ICML)*, 2017, p. 1183–1192.

[Gale, 1962] D. Gale et L. Shapley, « College admissions and the stability of marriage », *American Mathematical Monthly*, 1962.

[Gao, 2019a] J. Gao, J. Wang, S. Dai, L.-J. Li et R. Nevatia, « Note-rcnn : noise tolerant ensemble rcnn for semi-supervised object detection », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Gao, 2019b] Y. Gao, B. Liu, N. Guo, X. Ye, F. Wan, H. You et D. Fan, « C-midn : coupled multiple instance detection network with segmentation guidance for weakly supervised object detection », *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[Geifman, 2017] Y. Geifman et R. El-Yaniv, « Deep active learning over the long tail », *ArXiv*, t. abs/1711.00941, 2017.

[Geman, 1984] S. Geman et D. Geman, « Stochastic relaxation, gibbs distribution, and the bayesian restoration of images », *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, t. 6, n° 6, p. 721–741, 1984.

[Gidaris, 2018a] S. Gidaris, P. Singh et N. Komodakis, « Unsupervised representation learning by predicting image rotations », *International Conference on Representation Learning (ICLR)*, 2018.

[Gidaris, 2021] S. Gidaris, A. Bursuc, G. Puy, N. Komodakis, M. Cord et P. Pérez, « Online bag-of-visual-words generation for unsupervised representation learning », *CVPR*, 2021.

[Gidaris, 2020] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez et M. Cord, « Learning representations by predicting bags of visual words », *CVPR*, 2020.

[Gidaris, 2015] S. Gidaris et N. Komodakis, « Object detection via a multi-region and semantic segmentation-aware cnn model », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[Gidaris, 2018b] S. Gidaris, P. Singh et N. Komodakis, « Unsupervised representation learning by predicting image rotations », *ICLR*, 2018.

[Girshick, 2015] R. Girshick, « Fast R-CNN », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[Girshick, 2014] R. Girshick, J. Donahue, T. Darrell et J. Malik, « Rich feature hierarchies for accurate object detection and semantic segmentation », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Gissin, 2019] D. Gissin et S. Shalev-Shwartz, « Discriminative active learning », *ArXiv*, t. abs/1907.06347, 2019.

[Grauman, 2006] K. Grauman et T. Darrell, « Unsupervised learning of categories from sets of partially matching image features », *CVPR*, 2006.

[Greff, 2019] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick et A. Lerchner, « Multi-object representation learning with iterative variational inference », *ICML*, 2019.

[Grill, 2020] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar et al., « Bootstrap your own latent : a new approach to self-supervised learning », *NeurIPS*, 2020.

[Guo, 2019] P. Guo et R. Farrell, « Aligned to the object, not to the image : a unified pose-aligned representation for fine-grained recognition », *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, p. 1876–1885.

[Hara, 2018] K. Hara, H. Kataoka et Y. Satoh, « Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet ? », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Hariharan, 2012] B. Hariharan, J. Malik et D. Ramanan, « Discriminative decorrelation for clustering and classification », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[Haussmann, 2020] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet et J. M. Alvarez, « Scalable active learning for object detection », *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2020.

[He, 2021] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang et A. Yuille, « Transfg : a transformer architecture for fine-grained recognition », *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[He, 2020] K. He, H. Fan, Y. Wu, S. Xie et R. Girshick, « Momentum contrast for unsupervised visual representation learning », *CVPR*, 2020.

[He, 2017] K. He, G. Gkioxari, P. Dollar et R. Girshick, « Mask R-CNN », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[He, 2016] K. He, X. Zhang, S. Ren et J. Sun, « Deep residual learning for image recognition », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Hénaff, 2022] O. J. Hénaff, S. Koppula, E. Shelhamer, D. Zoran, A. Jaegle, A. Zisserman, J. Carreira et R. Arandjelović, *Object discovery and representation networks*, 2022.

[Hinton, 1977] G. E. Hinton, « Relaxation and its role in vision », thèse de doct., University of Edinburgh, 1977.

[Hjelm, 2019] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler et Y. Bengio, « Learning deep representations by mutual information estimation and maximization », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[Hoare, 1961] C. A. R. Hoare, « Algorithm 64 : quicksort », *Communications of the ACM*, t. 4, n° 7, 1961.

[Hochbaum, 2009] D. S. Hochbaum et V. Singh, « An efficient algorithm for co-segmentation », *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009, p. 269–276.

[Hsu, 2018a] K. Hsu, Y. Lin et Y. Chuang, « Co-attention cnns for unsupervised object co-segmentation », *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, p. 748–756.

[Hsu, 2019] K.-J. Hsu, Y.-Y. Lin et Y.-Y. Chuang, « Deepco$^3$ : deep instance co-segmentation by co-peak search and co-saliency detection », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Hsu, 2018b] K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, X. Qian et Y.-Y. Chuang, « Unsupervised cnn-based co-saliency detection with graphical optimization », *ECCV*, 2018.

[Hsu, 2015] W.-N. Hsu et H.-T. Lin, « Active learning by learning », 1, t. 29, 2015.

[Hu, 2018] H. Hu, J. Gu, Z. Zhang, J. Dai et Y. Wei, « Relation networks for object detection », *CVPR*, 2018.

[Huang, 2017] G. Huang, Z. Liu, L. van der Maaten et K. Q. Weinberger, « Densely connected convolutional networks », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Huang, 2014] S.-J. Huang, R. Jin et Z.-H. Zhou, « Active learning by querying informative and representative examples », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1936–1949, 2014.

[Huang, 2021] S. Huang, T. Wang, H. Xiong, J. Huan et D. Dou, « Semi-supervised active learning with temporal output discrepancy », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[Huang, 2020] Z. Huang, Y. Zou, B. Kumar et D. Huang, « Comprehensive attention self-distillation for weakly-supervised object detection », *Advances in Neural Information Processing Systems*, t. 33, 2020.

[Huang, 2019] Z. Huang, L. Huang, Y. Gong, C. Huang et X. Wang, « Mask scoring r-cnn », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, p. 6402–6411.

[Huffman, 1971] D. A. Huffman, « Impossible objects and nonsense sentences », *Machine Intelligence*, p. 295–323, 1971.

[Jeong, 2019] J. Jeong, S. Lee, J. Kim et N. Kwak, « Consistency-based semi-supervised learning for object detection », *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[Jerripothula, 2016] K. R. Jerripothula, J. Cai et J. Yuan, « Image co-segmentation via saliency co-fusion », *IEEE Transactions on Multimedia*, t. 18, n° 9, p. 1896–1909, 2016.

[Ji, 2019] X. Ji, J. F. Henriques et A. Vedaldi, « Invariant information clustering for unsupervised image classification and segmentation », *ICCV*, 2019.

[Jie, 2017a] Z. Jie, Y. Wei, X. Jin, J. Feng et W. Liu, « Deep self-taught learning for weakly supervised object localization », *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Jie, 2017b] Z. Jie, Y. Wei, X. Jin, J. Feng et W. Liu, « Deep self-taught learning for weakly supervised object localization », *CVPR*, 2017.

[Jing, 2008] Y. Jing et S. Baluja, « Visualrank : applying Pagerank to large-scale image search », *IEEE Trans. Pattern Anal. Machine Intell.*, 2008.

[Joulin, 2010] A. Joulin, F. Bach et J. Ponce, « Discriminative clustering for image co-segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[Joulin, 2014] A. Joulin, K. Tang et L. Fei-Fei, « Efficient image and video co-localization with Frank-Wolfe algorithm », *ECCV*, 2014.

[Kanade, 1980] T. Kanade, « A theory of the origami world », *Artificial Intelligence*, p. 279–311, 1980.

[Kang, 2019] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng et T. Darrell, « Few-shot object detection via feature reweighting », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, p. 8420–8429.

[Kao, 2018] C.-C. Kao, T.-Y. Lee, P. Sen et M.-Y. Liu, « Localization-aware active learning for object detection », *ACCV*, 2018.

[Karlinsky, 2019] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes et A. M. Bronstein, « Repmet : representative-based metric learning for classification and few-shot object detection », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, p. 5197–5206.

[Kass, 1988] M. Kass, A. Witkin et D. Terzopoulos, « Snakes : active contour models », *International Journal of Computer Vision (IJCV)*, t. 1, n° 4, p. 321–331, 1988.

[Katz, 1953] L. Katz, « A new status index derived from sociometric analysis », *Psychometrika*, t. 18, p. 39–43, 1953.

[Kelley, 1960] H. J. Kelley, « Gradient theory of optimal flight paths », *American Rocket Society*, t. 30, n° 10, p. 947–954, 1960.

[Kim, 2008] G. Kim, C. Faloutsos et M. Hebert, « Unsupervised modeling of object categories using link analysis techniques », *CVPR*, 2008.

[Kim, 2009] G. Kim et A. Torralba, « Unsupervised detection of regions of interest using iterative link analysis », *NIPS*, 2009.

[Kim, 2011] G. Kim et E. Xing, « Distributed cosegmentation via submodular optimization on anisotropic diffusion », *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[Kim, 2012] G. Kim et E. P. Xing, « On multiple foreground cosegmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[Kingma, 2014] D. P. Kingma, S. Mohamed, D. J. Rezende et M. Welling, « Semi-supervised learning with deep generative models », *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[Kleinberg, 1999] J. M. Kleinberg, « Authoritative sources in a hyperlinked environment », *JACM*, 1999.

[Koch, 2015] G. Koch, R. Zemel et R. Salakhutdinov, « Siamese neural networks for one-shot image recognition », *International Conference on Machine Learning Deep Learning Workshop*, 2015.

[Kolesnikov, 2016] A. Kolesnikov et C. H. Lampert, « Seed, expand and constrain : three principles for weakly-supervised image segmentation », *European Conference on Computer Vision (ECCV)*, 2016.

[Krasin, 2017] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan et K. Murphy, « Openimages : a public dataset for large-scale multi-label and multi-class image classification. », *Dataset available from https ://github.com/openimages*, 2017.

[Krizhevsky, 2012] A. Krizhevsky, I. Sutskever et G. E. Hinton, « Imagenet classification with deep convolutional neural networks », *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[Kuhn, 1955] H. W. Kuhn, « The hungarian method for the assignment problem », *Naval research logistics quarterly*, t. 2, p. 83–97, 1955.

[Kumar, 2010] M. Kumar, B. Packer et D. Koller, « Self-paced learning for latent variable models », *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[Kwak, 2015] S. Kwak, M. Cho, I. Laptev, J. Ponce et C. Schmid, « Unsupervised object discovery and tracking in video collections », *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[Laine, 2017] S. Laine et T. Aila, « Temporal ensembling for semi-supervised learning », *Proceedings of the International Conference in Learning Rrepresentations (ICLR)*, 2017.

[Landau, 1895] E. Landau, « Zur relativen wertbemessung der turnierresultate », *Deutsches Wochenschach*, t. 11, p. 366–369, 1895.

[Langville, 2004] A. Langville et C. Meyer, « Deeper inside Pagerank », *Internet Mathematics*, 2004.

[Langville, 2011] ——, *Google's PageRank and beyond : The science of search engine rankings*. 2011.

[Lazebnik, 2006] S. Lazebnik, C. Schmid et J. Ponce, « Beyond bags of features : spatial pyramid matching for recognizing natural scene categories », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[LeCun, 1989] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et L. D. Jackel, « Backpropagation applied to handwritten zip code recognition », *AT&T Bell Laboratories*, 1989.

[Lecun, 1998] Y. Lecun, L. Bottou, Y. Bengio et P. Haffner, « Gradient-based learning applied to document recognition », *Proceedings of the IEEE*, t. 86, n° 11, p. 2278–2324, 1998.

[Lee, 2015] C. Lee, W.-D. Jang, J.-Y. Sim et C.-S. Kim, « Multiple random walkers and their application to image cosegmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

References


[Lee, 2019] J. Lee, E. Kim, S. Lee, J. Lee et S. Yoon, « Ficklenet : weakly and semi-supervised semantic image segmentation using stochastic inference », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Lee, 2017] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr et M. Chandraker, « Desire : distant future prediction in dynamic scenes with interacting agents », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Lee, 2009] Y. J. Lee et K. Grauman, « Shape discovery from unlabeled image collections », *CVPR*, 2009.

[Lee, 2010] ——, « Object-graphs for context-aware category discovery », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[Li, 2019] B. Li, Z. Sun, Q. Li, Y. Wu et A. Hu, « Group-wise deep object co-segmentation with co-attention recurrent neural network », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[Li, 2015] C. Li, Y. Yuan, W. Cai, Y. Xia et D. Dagan Feng, « Robust saliency detection via regularized random walks ranking », *CVPR*, 2015.

[Li, 2020] Y. Li, D. Huang, D. Qin, L. Wang et B. Gong, « Improving object detection with selective self-supervised self-training », *ECCV*, 2020.

[Li, 2016] Y. Li, L. Liu, C. Shen et A. van den Hengel, « Image co-localization by mimicking a good detector's confidence score distribution », *ECCV*, 2016.

[Lin, 2017] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan et S. J. Belongie, « Feature pyramid networks for object detection », *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 936–944, 2017.

[Lin, 2014] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár et L. Zitnick, « Microsoft COCO : common objects in context », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[Lin, 2015] T.-Y. Lin, A. RoyChowdhury et S. Maji, « Bilinear cnn models for fine-grained visual recognition », *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, p. 1449–1457.

[Liu, 2021a] H. Liu, F. Wang, D. Guo, X. Liu, X. Zhang et F. Sun, « Active object discovery and localization using sound-induced attention », *IEEE Transactions on Industrial Informatics*, t. 17, n° 3, p. 2021–2029, 2021.

[Liu, 2015] S. Liu, J. Yang, C. Huang et M.-H. Yang, « Multi-objective convolutional learning for face labeling », *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, p. 3451–3459.

[Liu, 2021b] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin et B. Guo, « Swin transformer : hierarchical vision transformer using shifted windows », *ArXiv*, 2021.

[Liu, 2021c] Z. Liu, H. Ding, H. Zhong, W. Li, J. Dai et C. He, « Influence selection for active learning », *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, p. 9274–9283.

[Locatello, 2020] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy et T. Kipf, « Object-centric learning with slot attention », *NeurIPS*, 2020.

[Long, 2015] J. Long, E. Shelhamer et T. Darrell, « Fully convolutional networks for semantic segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, p. 3431–3440.

[Lowe, 2004] D. G. Lowe, « Distinctive image features from scale-invariant keypoints », *International Journal of Computer Vision (IJCV)*, 2004.

[Loy, 2013] C. C. Loy, C. Liu et S. Gong, « Person re-identification by manifold ranking », *ICIP*, 2013.

[M, 2020] G. M., Z. Z., Y. G., A. S.Ö., D. L.S. et P. T., « Consistency-based semi-supervised active learning : towards minimizing labeling cost », *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020.

[Mahoney, 2010] M. W. Mahoney et L. Orecchia, « Implementing regularization implicitly via approximate eigenvector computation », 2010.

[Manen, 2013] S. Manen, M. Guillaumin et L. Van Gool, « Prime object proposals with randomized prim's algorithm », *ICCV*, 2013.

[Mangalam, 2020] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik et A. Gaidon, « It is not the journey but the destination : endpoint conditioned trajectory prediction », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, p. 759–776.

[Marr, 1982] D. Marr, « Vision : a computational investigation into the human representation and processing of visual information », 1982.

[Matthies, 1989] L. Matthies, T. Kanade et R. Szeliski, « Kalman filter-based algorithms for estimating depth from image sequences », *International Journal of Computer Vision (IJCV)*, p. 209–236, 1989.

[Minderer, 2021] M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran et M. Lucic, « Revisiting the calibration of modern neural networks », *ArXiv*, 2021.

[Mises, 1929] R. V. Mises et H. Pollaczek-Geiringer, « Praktische verfahren der gleichungsauflösung. », *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 1929.

[Mishra, 2021] S. Mishra, A. Shah, A. Bansal, A. Jagannatha, A. Sharma, D. Jacobs et D. Krishnan, *Object-aware cropping for self-supervised learning*, 2021. arXiv : 2112.00319 [cs.CV].

[Miyato, 2019] T. Miyato, S.-I. Maeda, M. Koyama et S. Ishii, « Virtual adversarial training : a regularization method for supervised and semi-supervised learning », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. 41, n° 8, p. 1979–1993, 2019.

[Monnier, 2021] T. Monnier, E. Vincent, J. Ponce et M. Aubry, « Unsupervised layered image decomposition into object prototypes », *ArXiv*, 2021.

[Mukherjee, 2009] L. Mukherjee, V. Singh et C. R. Dyer, « Half-integrality based algorithms for cosegmentation of images », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, p. 2028–2035.

[Munkhdalai, 2017] T. Munkhdalai et H. Yu, « Meta networks », *Proceedings of the 34th International Conference on Machine Learning (ICML)*, t. 70, 2017, p. 2554–2563.

[Naseer, 2021] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan et M.-H. Yang, « Intriguing properties of vision transformers », *ArXiv*, 2021.

[Nedić, 2009] A. Nedić et A. Ozdaglar, « Approximate primal solutions and rate analysis for dual subgradient methods », *SIAM Journal on Optimization*, 2009.

[Nguyen, 2016] D. T. Nguyen, W. Li et P. O. Ogunbona, « Human detection from images and videos : a survey », *Pattern Recognition*, t. 51, p. 148–175, 2016.

[Noroozi, 2016] M. Noroozi et P. Favaro, « Unsupervised learning of visual representations by solving jigsaw puzzles », *European Conference in Computer Vision (ECCV)*, 2016, p. 69–84.

[Noroozi, 2017] M. Noroozi, H. Pirsiavash et P. Favaro, « Representation learning by learning to count », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[Oord, 2018] A. v. d. Oord, Y. Li et O. Vinyals, *Representation learning with contrastive predictive coding*, 2018.

[Ouaknine, 2021] A. Ouaknine, A. Newson, J. Rebut, F. Tupin et P. Pérez, « Carrada dataset : camera and automotive radar with range- angle- doppler annotations », *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, p. 5068–5075.

[Oudot, 2015] S. Oudot, « Persistence theory : from quiver representations to data analysis », *AMS Surveys and Monographs*, 2015.

[Page, 1999] L. Page, S. Brin, R. Motwani et T. Winograd, « The Pagerank citation ranking : bringing order to the web. », Stanford InfoLab, rapp. tech., 1999.

[Pan, 2019] T. Pan, B. Wang, G. Ding, J. Han et J. Yong, « Low shot box correction for weakly supervised object detection », *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, p. 890–896.

[Pardo, 2021] A. Pardo, M. Xu, A. K. Thabet, P. Arbeláez et B. Ghanem, « Baod : budget-aware object detection », *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 1247–1256, 2021.

[Pathak, 2016] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell et A. A. Efros, « Context encoders : feature learning by inpainting », *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, p. 2536–2544.

[Payandeh, 2019] S. Payandeh et E. Chiu, « Application of modified Pagerank algorithm for anomaly detection in movements of older adults », *IJTA*, 2019.

[Perron, 1907] O. Perron, « Grundlagen für eine theorie des jacobischen kettenbruchalgorithmus », 1907.

[Perronnin, 2007] F. Perronnin et C. Dance, « Fisher kernels on visual vocabularies for image categorization », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[Pinski, 1976] G. Pinski et F. Narin, « Citation influence for journal aggregates of scientific publications : theory, with application to the literature of physics », *Information Processing & Management*, t. 12, p. 297–312, 1976.

[Poggio, 1985] T. Poggio, V. Torre et C. Koch, « Computational vision and regularization theory », *Nature*, t. 317, n° 6035, p. 314–319, 1985.

[Qiao, 2018] S. Qiao, C. Liu, W. Shen et A. L. Yuille, « Few-shot image recognition by predicting parameters from activations », *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Quan, 2016a] R. Quan, J. Han, D. Zhang et F. Nie, « Object co-segmentation via graph optimized-flexible manifold ranking », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Quan, 2016b] ——, « Object co-segmentation via graph optimized-flexible manifold ranking », *CVPR*, 2016.

[Radford, 2021] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger et I. Sutskever, « Learning transferable visual models from natural language supervision », *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021, p. 8748–8763.

[Radosavovic, 2018] I. Radosavovic, P. Dollár, R. B. Girshick, G. Gkioxari et K. He, « Data distillation : towards omni-supervised learning », *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4119–4128, 2018.

[Radosavovic, 2017] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari et K. He, « Data distillation : towards omni-supervised learning », *CVPR*, 2017.

[Rahmani, 2008] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek et J. E. Fritts, « Localized content-based image retrieval », *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008.

[Ramachandran, 2019] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya et J. Shlens, « Stand-alone self-attention in vision models », *NeurIPS*, 2019.

[Ravi, 2017] S. Ravi et H. Larochelle, « Optimization as a model for few-shot learning », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[Redmon, 2016] J. Redmon, S. Divvala, R. Girshick et A. Farhadi, « You only look once : unified, real-time object detection », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Redmon, 2017] J. Redmon et A. Farhadi, « Yolo9000 : better, faster, stronger », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Ren, 2018] Q. Ren et R. Hu, « Saliency detection via pageank and local spline regression », *Journal of Electronic Imaging*, 2018.

[Ren, 2015a] S. Ren, K. He, R. Girshick et J. Sun, « Faster R-CNN : towards real-time object detection with region proposal networks », *NeuRIPS*, 2015.

[Ren, 2015b] ——, « Faster r-cnn : towards real-time object detection with region proposal networks », *NeurIPS*, 2015.

[Ren, 2020a] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, Y. J. Lee, A. G. Schwing et J. Kautz, « Instance-aware, context-focused, and memory-efficient weakly supervised object detection », *CVPR*, 2020.

[Ren, 2020b] Z. Ren, Z. Yu, X. Yang, M.-Y. Liu, A. G. Schwing et J. Kautz, « UFO$^2$ : a unified framework towards omni-supervised object detection », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Roberts, 1963] L. G. Roberts, « Machine perception of three-dimensional solids », thèse de doct., Massachusetts Institute of Technology, 1963.

[Rosenblatt, 1958] F. Rosenblatt, « The perceptron : a probabilistic model for information storage and organization in the brain », *Psychological Review*, t. 65, n° 6, p. 386–408, 1958.

[Rosenfeld, 1976] A. Rosenfeld, R. A. Hummel et S. W. Zucker, « Scene labeling by relaxation operations », *IEEE Transactions on Systems, Man, and Cybernetics*, p. 420–433, 1976.

[Rother, 2006] C. Rother, T. Minka, A. Blake et V. Kolmogorov, « Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006, p. 993–1000.

[Roy, 2018] S. Roy, A. Unmesh et V. P. Namboodiri, « Deep active learning for object detection », *British Machine Vision Conference (BMVC)*, 2018.

[Rubinstein, 2013] M. Rubinstein et A. Joulin, « Unsupervised joint object discovery and segmentation in internet images », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[Rumelhart, 1986a] D. E. Rumelhart, G. E. Hinton et R. J. Williams, « Learning internal representations by error propagation », *MIT Press*, 1986.

[Rumelhart, 1986b] ——, « Learning representations by back-propagating errors », *Nature*, t. 323, n° 6088, p. 533–536, 1986.

[Russakovsky, 2015] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg et L. Fei-Fei, « ImageNet large scale visual recognition challenge », *International Journal of Computer Vision (IJCV)*, t. 115, n° 3, p. 211–252, 2015.

[Russell, 2006] B. Russell, W. Freeman, A. Efros, J. Sivic et A. Zisserman, « Using multiple segmentations to discover objects and their extent in image collections », *CVPR*, 2006.

[Sanchez, 2013] J. Sanchez, F. Perronnin, T. Mensink et J. Verbeek, « Image classification with the fisher vector : theory and practice », *International Journal of Computer Vision (IJCV)*, 2013.

[Selvaraju, 2017] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh et D. Batra, « Grad-cam : visual explanations from deep networks via gradient-based localization », *ICCV*, 2017.

[Sener, 2018] O. Sener et S. Savarese, « Active learning for convolutional neural networks : a core-set approach », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[Settles, 2009] B. Settles, « Active Learning Literature Survey », University of Wisconsin-Madison Department of Computer Sciences, Technical Report, 2009.

[Siméoni, 2021a] O. Siméoni, M. Budnik, Y. Avrithis et G. Gravier, « Rethinking deep active learning : using unlabeled data at model training », *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2021, p. 1220–1227.

[Siméoni, 2019] O. Siméoni, A. Iscen, G. Tolias, Y. Avrithis et O. Chum, « Graph-based particular object discovery », *Machine Vision and Applications*, 2019.

[Siméoni, 2021b] O. Siméoni, G. Puy, H. V. Vo, S. Roburin, S. Gidaris, A. Bursuc, P. Pérez, R. Marlet et J. Ponce, « Localizing objects with self-supervised transformers and no labels », *Proceedings of the British Machine Vision Conference (BMVC)*, 2021.

[Simonyan, 2015a] K. Simonyan et A. Zisserman, « Very deep convolutional networks for large-scale image recognition », *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[Simonyan, 2015b] ——, « Very deep convolutional networks for large-scale image recognition », *International Conference on Learning Representations*, 2015.

[Sivic, 2008] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman et A. A. Efros, « Unsupervised discovery of visual object class hierarchies », *CVPR*, 2008.

[Sivic, 2005] J. Sivic, B. Russell, A. Efros, A. Zisserman et W. Freeman, « Discovering objects and their location in images », *ICCV*, 2005.

[Slater, 1950] M. Slater, « Lagrange multipliers revisited », *Cowles Commission Discussion Paper No. 403*, 1950.

[Snell, 2017] J. Snell, K. Swersky et R. S. Zemel, « Prototypical networks for few-shot learning », *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, p. 4080–4090.

[Sohn, 2020a] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang et C. Raffel, « Fixmatch : simplifying semi-supervised learning with consistency and confidence », *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[Sohn, 2020b] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee et T. Pfister, « A simple semi-supervised learning framework for object detection », *ArXiv :2005.04757*, 2020.

[Song, 2014a] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui et T. Darrell, *On learning to localize objects with minimal supervision*, 2014.

[Song, 2014b] H. O. Song, Y. J. Lee, S. Jegelka et T. Darrell, « Weakly-supervised discovery of visual pattern configurations », *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[Song, 2019] H. Song, H. Liang, H. Li, Z. Dai et X. Yun, « Vision-based vehicle detection and counting system using deep learning in highway scenes », *European Transport Research Review*, n° 51, 2019.

[Strudel, 2021] R. Strudel, R. Garcia, I. Laptev et C. Schmid, « Segmenter : transformer for semantic segmentation », *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[Sun, 2021] B. Sun, B. Li, S. Cai, Y. Yuan et C. Zhang, « Fsce : few-shot object detection via contrastive proposal encoding », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, p. 7352–7362.

[Sung, 2018] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr et T. M. Hospedales, « Learning to compare : relation network for few-shot learning », *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, p. 1199–1208.

[Szeliski, 2010] R. Szeliski, *Computer Vision : Algorithms and Applications*. Springer, 2010.

[Tan, 2019] M. Tan et Q. Le, « EfficientNet : rethinking model scaling for convolutional neural networks », *Proceedings of the 36th International Conference on Machine Learning (ICML)*, t. 97, 2019, p. 6105–6114.

[Tang, 2008] J. Tang et P. H. Lewis, « Non-negative matrix factorisation for object class discovery and image auto-annotation », *CIVR*, 2008.

[Tang, 2014] K. Tang, A. Joulin et L.-j. Li, « Co-localization in real-world images », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[Tang, 2018a] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu et A. Yuille, « Pcl : proposal cluster learning for weakly supervised object detection », *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, t. 42, n° 1, p. 176–191, 2018.

[Tang, 2017] P. Tang, X. Wang, X. Bai et W. Liu, « Multiple instance detection network with online instance classifier refinement », *CVPR*, 2017.

[Tang, 2018b] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang et A. Yuille, « Weakly supervised region proposal network and object detection », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[Tang, 2021a] P. Tang, C. Ramaiah, Y. Wang, R. Xu et C. Xiong, « Proposal learning for semi-supervised object detection », *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[Tang, 2021b] P. Tang, C. Ramaiah, R. Xu et C. Xiong, « Proposal learning for semi-supervised object detection », 2021, p. 2290–2300.

[Tang, 2018c] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu et A. Yuille, « Pcl : proposal cluster learning for weakly supervised object detection », *TPAMI*, t. 42, 2018.

[Tarvainen, 2017] A. Tarvainen et H. Valpola, « Mean teachers are better role models : weight-averaged consistency targets improve semi-supervised deep learning results », *Advances in Neural Information Processing Systems (NeurIPS)*, t. 30, 2017.

[Terzopoulos, 1983] D. Terzopoulos, « Multilevel computational processes for visual surface reconstruction », *Computer Vision, Graphics, and Image Processing*, t. 24, p. 52–96, 1983.

[Tian, 2021] H. Tian, Y. Chen, J. Dai, Z. Zhang et X. Zhu, « Unsupervised object detection with lidar clues », *CVPR*, 2021.

[Tian, 2020] Y. Tian, D. Krishnan et P. Isola, « Contrastive multiview coding », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Torralba, 2008] A. Torralba, R. Fergus et Y. Weiss, « Small codes and large image databases for recognition », *CVPR*, 2008.

[Touvron, 2020] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles et H. Jégou, « Training data-efficient image transformers & distillation through attention », *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2020.

[Touvron, 2021] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve et H. Jégou, « Going deeper with image transformers », *ArXiv*, 2021.

[Tuli, 2021] S. Tuli, I. Dasgupta, E. Grant et T. L. Griffiths, « Are convolutional neural networks or transformers more like human vision ? », *CogSci*, 2021.

[Tuytelaars, 2010] T. Tuytelaars, C. Lampert, M. Blaschko et W. Buntine, « Unsupervised object discovery : a comparison », *Int. Journal on Computer Vision*, 2010.

[Uijlings, 2013] J. Uijlings, K. van de Sande, T. Gevers et A. Smeulders, « Selective search for object recognition », *International Journal on Computer Vision*, 2013.

[Vaswani, 2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser et I. Polosukhin, « Attention is all you need », *NeurIPS*, 2017.

[Vernaza, 2017] P. Vernaza et M. Chandraker, « Learning random-walk label propagation for weakly-supervised semantic segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Vinyals, 2016] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu et D. Wierstra, « Matching networks for one shot learning », *Neural Information Processing Systems (NeurIPS)*, 2016.

[Vo, 2019] H. V. Vo, F. Bach, M. Cho, K. Han, Y. LeCun, P. Pérez et J. Ponce, « Unsupervised image matching and object discovery as optimization », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Vo, 2020] H. V. Vo, P. Pérez et J. Ponce, « Toward unsupervised, multi-object discovery in large-scale image collections », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Vo, 2022] H. V. Vo, O. Siméoni, S. Gidaris, A. Bursuc, P. Pérez et J. Ponce, « Active learning strategies for weakly-supervised object detection », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[Vo, 2021] H. V. Vo, E. Sizikova, C. Schmid, P. Pérez et J. Ponce, « Large-scale unsupervised object discovery », *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Vu, 2021] T. Vu, K. Haeyong et C. D. Yoo, « Scnet : training inference sample consistency for instance segmentation », *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[Vu, 2019] T. Vu, H. Jain, M. Bucher, M. Cord et P. Perez, « Advent : adversarial entropy minimization for domain adaptation in semantic segmentation », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Waltz, 1975] D. L. Waltz, « Understanding line drawings of scenes with shadows », *The Psychology of Computer Vision*, 1975.

[Wan, 2019] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao et Q. Ye, « C-mil : continuation multiple instance learning for weakly supervised object detection », *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[Wang, 2013] H. Wang et C. Schmid, « Action recognition with improved trajectories », *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013, p. 3551–3558.

[Wang, 2018a] K. Wang, X. Yan, D. Zhang, L. Zhang et L. Lin, « Towards human-machine cooperation : self-supervised sample mining for object detection », *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Wang, 2016] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang et L. Val Gool, « Temporal segment networks : towards good practices for deep action recognition », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[Wang, 2018b] X. Wang, R. Girshick, A. Gupta et K. He, « Non-local neural networks », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Wang, 2018c] ——, « Non-local neural networks », *CVPR*, 2018.

[Wang, 2020a] X. Wang, R. Zhang, T. Kong, L. Li et C. Shen, « Solov2 : dynamic and fast instance segmentation », *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[Wang, 2018d] Y.-X. Wang, R. Girshick, M. Hebert et B. Hariharan, « Low-shot learning from imaginary data », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, p. 7278–7286.

[Wang, 2020b] Y. Wang, J. Zhang, M. Kan, S. Shan et X. Chen, « Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation », *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[Wang, 2020c] Z. Wang, L. Zheng, Y. Liu et S. Wang, « Towards real-time multi-object tracking », *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Weber, 2000] M. Weber, M. Welling et P. Perona, « Towards automatic discovery of object categories », *CVPR*, 2000.

[Wei, 2017a] X.-S. Wei, C.-L. Zhang, Y. Li, C.-W. Xie, J. Wu, C. Shen et Z.-H. Zhou, « Deep descriptor transforming for image co-localization », *IJCAI*, 2017.

[Wei, 2019] X.-S. Wei, C.-L. Zhang, J. Wu, C. Shen et Z.-H. Zhou, « Unsupervised object discovery and co-localization by deep descriptor transforming », *Pattern Recognition*, 2019.

[Wei, 2017b] X.-S. Wei, J.-H. Luo, J. Wu et Z.-H. Zhou, « Selective convolutional descriptor aggregation for fine-grained image retrieval », *IEEE Transactions on Image Processing*, t. 26, p. 2868–2881, 2017.

[Wei, 2018] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng et T. S. Huang, « Revisiting dilated convolution : a simple approach for weakly- and semi-supervised semantic segmentation », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Wojke, 2017] N. Wojke, A. Bewley et D. Paulus, « Simple online and realtime tracking with a deep association metric », *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2017, p. 3645–3649.

[Wu, 2019] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo et R. Girshick, *Detectron2*, https://github.com/facebookresearch/detectron2, 2019.

[Xie, 2021a] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez et P. Luo, « Segformer : simple and efficient design for semantic segmentation with transformers », *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[Xie, 2020] Q. Xie, Z. Dai, E. Hovy, T. Luong et Q. V. Le, « Unsupervised data augmentation for consistency training », *Advances in Neural Information Processing Systems (NeurIPS)*, t. 33, 2020.

[Xie, 2021b] Z. Xie, Y. Lin, Z. Yao, Z. Zhang, Q. Dai, Y. Cao et H. Hu, « Self-supervised learning with swin transformers », *ArXiv*, 2021.

[Xu, 2021] M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai et Z. Liu, « End-to-end semi-supervised object detection with soft teacher », 2021.

[Yang, 2021] Z. Yang, M. Shi, C. Xu, V. Ferrari et Y. Avrithis, « Training object detectors from few weakly-labeled and many unlabeled images », *Pattern Recognition*, p. 108 164, 2021.

[Yao, 2016] T. Yao, T. Mei et Y. Rui, « Highlight detection with pairwise deep ranking for first-person video summarization », *CVPR*, 2016.

[Yoo, 2019] D. Yoo et I. S. Kweon, « Learning loss for active learning », *CVPR*, 2019.

[Yuan, 2021a] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng et S. Yan, « Tokens-to-token vit : training vision transformers from scratch on imagenet », *ArXiv*, 2021.

[Yuan, 2021b] T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji et Q. Ye, « Multiple instance active learning for object detection », *CVPR*, 2021.

[Zbontar, 2021] J. Zbontar, L. Jing, I. Misra, Y. LeCun et S. Deny, « Barlow twins : self-supervised learning via redundancy reduction », *International Conference on Machine Learning (ICML)*, 2021.

[Zeiler, 2014] M. D. Zeiler et R. Fergus, « Visualizing and understanding convolutional networks », *ECCV*, 2014.

[Zeng, 2019] Z. Zeng, B. Liu, J. Fu, H. Chao et L. Zhang, « Wsod2 : learning bottom-up and top-down objectness distillation for weakly-supervised object detection », *Proceedings of the IEEE International Conference on Computer Vision (ICV)*, 2019.

[Zhai, 2021] X. Zhai, A. Kolesnikov, N. Houlsby et L. Beyer, « Scaling vision transformers », *ArXiv*, 2021.

[Zhang, 2020a] B. Zhang, L. Li, S. Yang, S. Wang, Z. Zha et Q. Huang, « State-relabeling adversarial active learning », *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 8753–8762, 2020.

[Zhang, 2020b] C.-L. Zhang, Y.-H. Cao et J. Wu, « Rethinking the route towards weakly supervised object localization », *CVPR*, 2020.

[Zhang, 2011] D. Zhang, F. Wang, L. Si et T. Li, « Maximum margin multiple instance clustering with applications to image and text clustering », *Transactions on Neural Networks*, 2011.

[Zhang, 2015a] D. Zhang, D. Meng, C. Li, L. Jiang, Q. Zhao et J. Han, « A self-paced multiple-instance learning framework for co-saliency detection », *ICCV*, 2015.

[Zhang, 2020c] D. Zhang, H. Zhang, J. Tang, X.-S. Hua et Q. Sun, « Causal intervention for weakly-supervised semantic segmentation », *Advances in Neural Information Processing Systems (NeurIPS)*, t. 33, 2020, p. 655–666.

[Zhang, 2009] M.-L. Zhang et Z.-H. Zhou, « Multi-instance clustering with applications to multi-instance prediction », *Applied Intelligence*, 2009.

[Zhang, 2015b] Q. Zhang, Y. Nian Wu et S.-C. Zhu, « Mining and-or graphs for graph matching and object discovery », *ICCV*, 2015.

[Zhang, 2016] R. Zhang, P. Isola et A. A. Efros, « Colorful image colorization », *European Conference on Computer Vision (ECCV)*, 2016, p. 649–666.

[Zhang, 2017] R. Zhang, P. Isola et A. A. Efros, « Split-brain autoencoders : unsupervised learning by cross-channel prediction », *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Zhang, 2020d] R. Zhang, Y. Huang, M. Pu, J. Zhang, Q. Guan, Q. Zou et H. Ling, « Object discovery from a single unlabeled image by mining frequent itemsets with multi-scale features », *TIP*, t. 29, 2020.

[Zhang, 2018a] X. Zhang, Y. Wei, J. Feng, Y. Yang et T. S. Huang, « Adversarial complementary learning for weakly supervised object localization », *CVPR*, 2018.

[Zhang, 2018b] X. Zhang, Y. Wei, G. Kang, Y. Yang et T. Huang, « Self-produced guidance for weakly-supervised object localization », *ECCV*, 2018.

[Zhang, 2020e] X. Zhang, Y. Wei et Y. Yang, « Inter-image communication for weakly supervised localization », *ECCV*, 2020.

[Zhang, 2021] Y. Zhang, C. Wang, X. Wang, W. Zeng et W. Liu, « Fairmot : on the fairness of detection and re-identification in multiple object tracking », *International Journal of Computer Vision (IJCV)*, t. 129, p. 3069–3087, 2021.

[Zhdanov, 2019] F. Zhdanov, « Diverse mini-batch active learning », *ArXiv*, t. abs/1901.05954, 2019.

[Zhou, 2015] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva et A. Torralba, « Object detectors emerge in deep scene cnns », *ICLR*, 2015.

[Zhou, 2016] ——, « Learning deep features for discriminative localization », *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[Zhu, 2012] J.-Y. Zhu, J. Wu, Y. Xu, E. Chang et Z. Tu, « Unsupervised object class discovery via saliency-guided multiple class learning », *CVPR*, 2012.

[Zhu, 2021] X. Zhu, W. Su, L. Lu, B. Li, X. Wang et J. Dai, « Deformable {detr} : deformable transformers for end-to-end object detection », *ICLR*, 2021.

[Zhuang, 2020] P. Zhuang, Y. Wang et Y. Qiao, « Learning attentive pairwise interaction for fine-grained classification », *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, p. 13 130–13 137.

[Zitnick, 2014] L. Zitnick et P. Dollár, « Edge boxes : locating object proposals from edges », *European Conference on Computer Vision*, 2014.

[Zomorodian, 2005] A. Zomorodian et G. Carlsson, « Computing persistent homology », *Discrete and Computational Geometry*, 2005.

[Zoph, 2020] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk et Q. V. Le, « Rethinking pre-training and self-training », *NeurIPS*, 2020.

## RÉSUMÉ

Les modèles de détection d'objets dans les images sont des composants importants de systèmes intelligents comme les véhicules autonomes ou les robots. Ils sont typiquement obtenus par l'apprentissage supervisé, ce qui nécessite de grands jeux de données annotées à la main. La construction de tels jeux de données est pourtant coûteuse en temps et en argent, ce qui limite souvent leur taille et leur diversité et, par conséquent, restreint l'applicabilité des détecteurs d'objets. L'objectif de cette thèse est de développer des alternatives à l'apprentissage supervisé qui demandent moins de données annotées pour la détection d'objets. Dans la première partie de la thèse, nous nous concentrons sur la découverte d'objets non-supervisée, qui, étant donné une collection d'images non-annotées, vise à trouver les images qui contiennent les objets de la même catégorie, et localiser ces objets. Nous introduisons deux méthodes d'optimisation discrète (OSD et rOSD), une méthode de classement (LOD) et une méthode qui se base sur les descripteurs des transformers auto-supervisés pour ce problème. Dans la deuxième partie de la thèse, nous considérons un scénario pratique qui combine l'apprentissage faible et actif pour entraîner un détecteur d'objets, et discutons BiB, une méthode efficace pour un tel scénario. Nous démontrons que BiB offre un meilleur compromis entre la performance de détection et le coût d'annotation que l'apprentissage faiblement et complètement supervisé.

## MOTS CLÉS

découvert d'objets, détection d'objets, apprentissage non-supervisé, apprentissage actif, apprentissage faiblement supervisé, optimisation.

## ABSTRACT

Object detectors are important components of intelligent systems such as autonomous vehicles or robots. They are typically obtained with fully-supervised training, which requires large manually annotated datasets whose construction is time-consuming and costly. This thesis studies alternatives to fully-supervised object detection that work with less or even no manual annotation. We focus in the first part of this thesis on the unsupervised object discovery problem, which, given an image collection without manual annotation, aims at identifying pairs of images that contain similar objects and localizing these objects. We discuss two optimization-based approaches(OSD and rOSD), a ranking method (LOD) and a simple seed-growing approach that exploits features from self-supervised transformers (LOST) to this problem. In the second part of the thesis, we consider a practical scenario which combines weakly-supervised and active learning for training an object detector, and propose BiB, an active learning strategy tailored for this scenario. We show that our pipeline offers a better trade-off between annotation cost and effectiveness than both weakly- and fully-supervised object detection.

## KEYWORDS

object discovery, object detection, unsupervised learning, weakly-supervised learning, active learning, optimization.