Learning to Discover and Detect Objects

Anonymous Author(s)
Affiliation
Address
email

Abstract

We tackle the problem of novel class discovery, detection, and localization (NCDL). In this setting, we assume a *source* dataset with labels for objects of commonly observed classes. Instances of other classes need to be discovered, classified, and localized automatically based on visual similarity, without human supervision. To this end, we propose a two-stage object detection network *Region-based NCDL* (*RNCDL*), that uses a region proposal network to localize potential objects and classify them. We train our network to classify each proposal, either as one of the *known* classes, seen in the source dataset, or one of the extended set of *novel* classes with a constraint that the distribution of class assignments should follow natural long-tail distributions common in the real open-world. By training our detection network with this objective in an end-to-end manner, it learns to classify all region proposals for a large variety of classes, including those that are not part of the labeled object class vocabulary. Our experiments conducted using COCO and LVIS datasets reveal that our method is significantly more effective compared to multi-stage pipelines that rely on traditional clustering algorithms or self-supervised contrastive learning methods and operate on pre-extracted crops. Beyond that, we demonstrate the generality of our approach by applying our method to a large-scale Visual Genome dataset, where our network successfully learns to detect various semantic classes without explicit supervision. Code is available at: `https://github.com/5da15f23f9/learning-to-discover-and-detect-objects`.

1 Introduction

We tackle novel class discovery and localization in unlabeled datasets, a long-standing problem in computer vision [55, 59, 41, 42, 54]. As we are approaching the limits of well-understood and successful supervised training of object detectors [22, 21, 53, 27], this is becoming an increasingly more important problem, with the task of labeling novel and rare classes that appear in the long tail of object class distribution becoming prohibitively expensive [24].

Object class discovery is a difficult problem, as there may be many possible equally valid attributes for grouping patterns, such as object color or orientation. As shown in [29, 30], *novel class discovery (NCD)* can be well-posed and tackled in a data-driven manner by injecting prior knowledge on *how we wish to group semantic classes* using some degree of supervision. Existing methods [29, 30, 26, 25, 72, 7, 73, 74, 19] all assume curated datasets, where objects of interest are pre-cropped and semantic classes are fairly balanced. In contrast, we tackle joint *novel class discovery and localization (NCDL)*, a task of learning to discover and detect objects from raw, unlabeled, and uncurated data. This is a significantly more challenging problem, as images always contain a mixture of *labeled* and *unlabeled* object classes, and we need to localize as well as categorize them.

To study this problem, we re-purpose the COCO [44] and LVIS [24] datasets. We use 50% of COCO images together with COCO class vocabulary and labels for supervised model training. We use the remaining 50% of images without any labels during the training and evaluate how well our network learns to detect novel classes using the long-tailed LVIS label set. This setting allows us to mimic

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

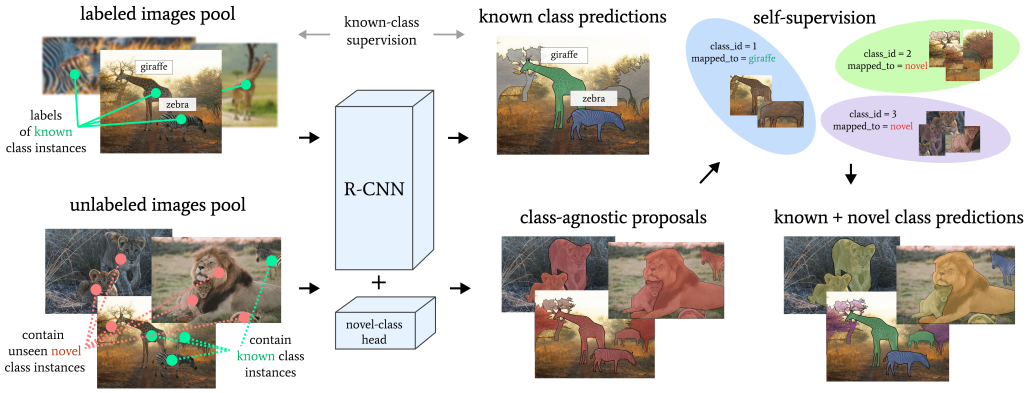

Figure 1: In NCDL problem we operate with labeled images pool containing annotations for *known*, frequently observed semantic classes, and unlabeled images pool that may contain instances of *novel* classes. Our network learns to localize and recognize common semantic classes, as well as categorize novel classes, for which supervision was not given.

real-world scenarios where no supervision is given for the target domain, while a rich set of labels allows us to rigorously study this problem.

We train a two-stage object detector [53, 27] in an end-to-end manner such that (i) we can correctly detect (*i.e.* localize and categorize) *labeled* objects, and, in parallel, (ii) detect *unlabeled* objects by learning feature representations suitable for categorization of objects that would otherwise be classified as the *background* class. We start with supervised training on the labeled, *source* domain, followed by the *self-supervised* training on the target (unlabeled) domain. We transfer knowledge from the source to the target domain by *retaining* weights for class-agnostic modules, such as RPN and bounding box regression, and instance segmentation heads. For classification, we reuse the primary classification head and add a new secondary head, training them jointly with an objective to categorize *every* region proposal consistently under multiple transformations and a constraint that the marginal distribution of class assignments follows a long-tail distribution. By encouraging the network to learn features for a variety of objects via a non-uniform classification prior, we also effectively reduce the bias towards simply classifying regions as one of the *labeled* or *background* classes.

Our end-to-end trainable method significantly outperforms prior art [68] that learns object representations using contrastive learning followed by k-means clustering, while providing categorization for *any* object with a single network forward pass. More importantly, with this work, we lay down the foundations for an exciting new line of research that goes beyond well-established supervised learning for object detection.

In summary, this paper makes the following **contributions**: (i) We study the problem of *novel class discovery* in the wild using standard object detection datasets that do not pre-localize regions of interest manually or assume objects are roughly centered, *i.e.*, photographer bias. (ii) We propose a simple yet effective and scalable method that can be trained end-to-end on the unlabeled domain. Our network is bootstrapped using a labeled dataset to learn features suitable for categorizing future instances. We train feature representations for arbitrary objects using the self-supervised objective. Finally (iii), we reveal that our method can discover and detect novel class instances and outperform prior work, demonstrating the generalization to datasets beyond COCO.

2 Related Work

Object discovery in image and video collections. Discovery of semantic objects that appear in unlabeled video collections is a long-standing research problem [60]. Early methods employ multiple bottom-up segmentations of images [55, 59] to discover commonly occurring patterns using topic modeling, *e.g.*, Dirichlet allocation [6]. Lee *et al.* propose to rely on saliency [41] or to incorporate semantic knowledge about certain semantic classes, for which supervision is available [42]. These methods are evaluated on smaller datasets [23, 58, 17] that contain one or a handful of well-delineated objects. Rubinstein *et al.* [54] demonstrated that discovery can be scaled to internet photo collections.

Beyond images, Kwak *et al.* [40] tackle the discovery of objects in YouTube videos that usually contain a single object exhibiting dominant motion. They build on a two-stage approach: (i) identify the most salient region proposals in terms of appearance and motion, and (ii) co-segment consistently appearing objects across video clips. Video object discovery videos, recorded from a moving platform, were tackled in [49, 50] by first mining video-object tracks by associating region proposal network (RPN) [53] based proposals across time, followed by clustering of video proposals based on pre-trained CNN features. Recent work on open-set panoptic segmentation [32] similarly employs clustering based on RPN features to discover commonly-occurring objects, used as pseudo-labels for learning to segment unlabeled objects. The method by [68] follows the general approach of clustering RPN proposals, however, instead of relying on pre-trained CNN features, they learn representations suitable for clustering using self-supervised objective, trained via contrastive learning.

While most of the foregoing works focus on saliency criteria to significantly narrow down the number of region proposals, [62, 63] resort to combinatorial optimization that can cope with a large set of object proposals to discover objects that frequently appear in image datasets. It has been shown [64] that this approach can be scaled to large image collections, *e.g.*, OpenImages [39].

Novel class discovery. In Novel class discovery (NCD) works, given is labeled data that can be used to train a feature extractor and learn a clustering criterion to guide a notion of similarity and unlabeled data is assumed to contain novel class instances only. Hsu *et al.* [29, 30] propose a framework that injects prior knowledge on how we wish to group semantic classes from supervised learning on *labeled* images to the *unlabeled* domain via learnable similarity function. The work of Han *et al.* [26] builds on deep clustering [71] and suggests an approach for automatically discovering the number of novel classes. In follow-up work, Han *et al.* [25] build on pairwise similarity of image features [12] and propose leveraging self-supervised pretraining along with freezing the backbone to reduce bias towards the known classes. Its follow-up works [72, 34, 74, 73] further boost the performance by introducing additional self-supervised and regularization components. UNO [19] proposes a simple method based on SwAV [11] that bootstraps signal for novel class training using pseudo-labels generated under equipartitioning constraints. ORCA [7] extends the NCD problem and operates under the assumption that unlabeled images can contain instances of known classes as well. Motivated by [61], [12], [65], authors leverage self-supervised initialization of backbone, supervised loss along with unsupervised loss based on pairwise pseudo-labels, and novel dynamic softmax margin [65] to alleviate bias of the known classes. In contrast to our work, the foregoing methods investigate the problem in the image classification setting, assuming objects of interest are localized (pre-cropped), and operate under absence of outliers (*e.g.*, images containing no objects).

Related and complementary research directions. Methods for *generic* [45] or *large vocabulary object detection* [24] focus on detecting/segmenting a large set of a-priori known semantic classes, often by utilizing multiple datasets and (possibly weak) supervision [75, 52, 31]. A special case is *zero-shot learning* [70] and *zero-shot object detection* [3, 48, 51], where (mostly weak) supervision for *unseen* classes comes in the form of attributes, class names, latent features, or other forms of auxiliary data, commonly used in a conjunction with language models embeddings to learn a joint embedding space. In *few-shot learning* [36, 66, 18, 8] given is a limited supervision, where algorithms are required to learn from one or a few data samples per each class. In [18, 8] authors employ a secondary classification head that is trained using few-shot supervision and distillation [18] or margin loss [8]. In contrast to zero-shot, few-shot, and large-vocabulary detection, we assume no supervision and prior knowledge for object classes that appear in the tail.

Methods for *open-set recognition* [56, 57, 33, 5] and *detection* [48, 16] focus on calibration of per-class uncertainties to minimize the confusion between known and possibly unknown classes. *Open-world recognition* methods, as defined by [4, 46], must explicitly recognize *unknown* object instances that were not observed as labeled samples during the model training, and must *continually* update object detectors to recognize these unknown instances. Recent method for *open-world* detection [35] focuses on the *continual learning* aspect of the task: for the evaluation, labels for "novel" classes are fully provided (those are the held-out classes). Our work is orthogonal, and tackles the "missing bit" of the open-world detection pipeline: discovery and semantic grouping of *novel*-class objects that *can appear* in our training data as well, however, are unlabeled. Clusters of discovered *novel* classes could be verified by human annotators and used to update detection models in a continual fashion, as proposed by [4].

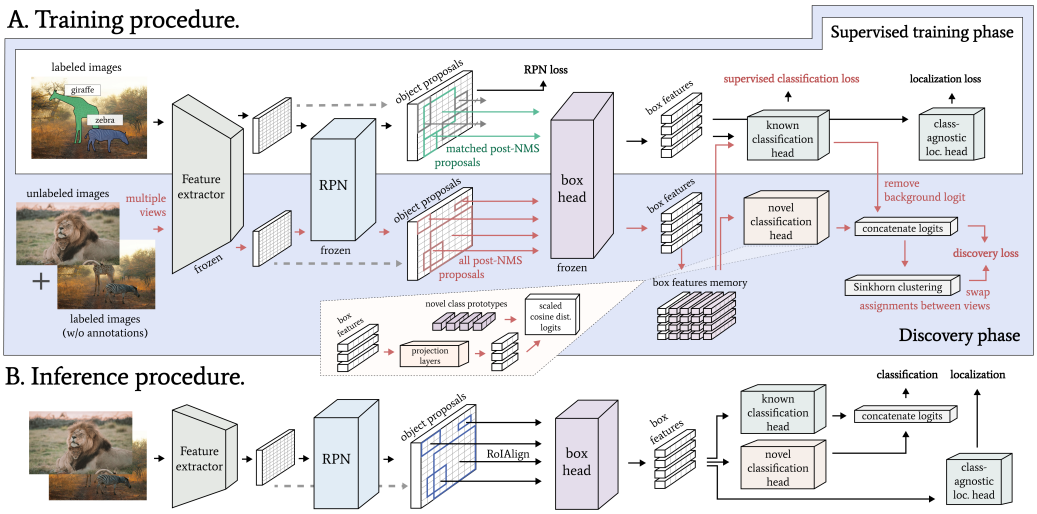

Figure 2: A high-level overview of our network. *(top)* ***During the supervised training phase***, we train our backbone and RPN networks, together with classification head for the *labeled* classes, a class agnostic localization head. ***During the discovery phase*** we freeze all the layers of the network apart from classification head and attach a novel classification head. ***During the inference*** *(bottom)* we perform a standard R-CNN pass, using classification heads of both known and novel categories to predict proposals classes.

3 Novel Class Discovery and Localization

We first detail *novel class discovery and localization (NCDL)*, the task of learning to discover and detect objects from raw, unlabeled, and uncurated data. Given is a source (*labeled*) object detection image dataset D^k for (*known*) object classes C^k ($|C^k| = K$), that provides both class and box level objects supervision: $D^k = \{(\mathbf{I}, \{(\mathbf{b}, c)\})\}$, where $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$, $\mathbf{b} \in \mathbb{R}^4$, and $c \in C^k$. In addition, given is one or more target (*unlabeled*) datasets $D^n = \{I\}$ that may contain instances of both previously seen and unseen (*novel*) object classes $C^k \cup C^n$ ($|C^n| = N, N > K$) that we need to learn to detect in the absence of any labels. For an arbitrary image I from the source or target domain, NCDL methods should detect all object instances present in the image: $\{(\mathbf{b}', c')\}$, *i.e.* localize them along with predicting their semantic class (*e.g.*, a car needs to be detected as the `car` class).

3.1 Region-based Novel Category Discovery and Localization (RNCDL)

As the main contribution of this paper, we propose an end-to-end trainable detection and discovery network that jointly learns discriminative representations for all region proposals, their localization, and categorization alleviating the need for explicit clustering after the network training. We build on a two-stage Faster or Mask R-CNN [53, 27]. Such detectors are naturally suitable for NCDL as their region proposal network (RPN) is trained in a class-agnostic manner to propose regions of interest that likely contain objects. Given object-centric feature representations for each localized candidate region, that we learn via supervised pre-training, one natural approach for discovering (categorizing) *novel* classes would be to cluster such regions in feature space using, *e.g.*, k-means, *c.f.*, [49, 32, 68]. However, leveraging feature representation learned purely via supervised R-CNN training is likely to not be discriminative *enough* with respect to objects of *unlabeled* classes. Moreover, such approach is prone to having a strong bias toward classifying novel instances as one of the *labeled* classes. To this end, we introduce our end-to-end RNCDL framework that employs pseudo-labeling obtained via constrained clustering. Via a newly attached classification head, it learns to discriminate objects of *all* classes, including the *unlabeled* ones, building on learned feture representations.

Supervised training on the source domain. We begin with a *self-supervised* backbone pre-training [28] on a large-scale image dataset, such as ImageNet [15], and use it to initialize the backbone for an R-CNN network. We then train the object detector on the source (labeled) domain using supervision for objects of *known* C^k classes, such as 80 COCO classes (Fig. 2A, *top*). We use the labeled data to train the shared feature extractor (backbone) together with a class-agnostic region proposal network (RPN), class-agnostic box regression head, as well as a classification head

for categorization of C^k classes. Specifically, for each image I, RPN is trained to predict a set of class-agnostic proposals $\{(\mathbf{b}, o)\}$, where $\mathbf{b} \in \mathbb{R}^4$ are predicted coordinates and $o \in \mathbb{R}$ is the objectness score. We then pass the proposals' feature representations, obtained from the backbone via the RoIAlign [27] mechanism, through an MLP (box head) and pass the resulting features $\hat{\mathbf{f}} \in \mathbb{R}^{\hat{F}}$ to RoI heads for the final predictions. We train a class-agnostic bounding box head to refine box location for each object $\hat{\mathbf{b}} = h^{bb}(\mathbf{b}, \hat{\mathbf{f}})$, and a classification box head to categorize each object into one of the K classes or a background: $\hat{c} = h^k(\hat{\mathbf{f}}) = \mathbf{W}^k\hat{\mathbf{f}}$, where $\hat{c} \in \mathbb{R}^{K+1}$ and $\mathbf{W}^k \in \mathbb{R}^{(K+1)\times\hat{F}}$. For training we use a standard fully-supervised R-CNN loss: $L_{sup} = L_{RPN} + L_{box} + L_{cls}$, consisting of the sum of RPN, localization, and classification losses respectively [53, 27].

Discovery on the target domain. During the following self-supervised (discovery) training phase on the target domain, we attach a secondary (novel) classification head h^n on top of RoI box features $\hat{\mathbf{f}}$ (Fig. 2B). The secondary classification head consists of an MLP and a linear classification layer that we detail below. We also modify the primary (known) classification head, by removing the *background* class from its weight matrix such that $\hat{\mathbf{W}}^k \in \mathbb{R}^{K\times\hat{F}}$. As the *background* class is removed at this phase, we aim to learn to cluster proposals. To obtain classification predictions at this phase, we pass RoI features through both h^k and h^n classification heads, concatenating their logits: $\hat{\mathbf{p}} = \text{softmax}([\mathbf{l}^k, \mathbf{l}^n]) = \text{softmax}([h^k(\hat{\mathbf{f}}), h^n(\hat{\mathbf{f}})]) \in \mathbb{R}^{K+N}$. During training, we freeze all layers except the classification heads. We train the classification heads *jointly* using a *self-supervised classification* loss and *supervised classification* loss: $L_{disc} = L_{ss} + \alpha \cdot L_{cls}$, where α is the supervised loss scale coefficient. We show in ablations that keeping the downscaled ($\alpha < 1$) supervised classification loss in the loop effectively alleviates forgetting the weights for the known classes, while large α may create an unwanted bias and harm learning of novel classes. As we remove the *background* class, different from the supervised phase, we calculate the supervised loss only for annotation-matched region proposals. For the *self-supervised classification loss* we use cross-entropy loss on top of pseudo-labels generated online for the current batch of proposals that we detail in a subsection below. The employed loss encourages the network to categorize every region proposal consistently under multiple transformations and a constraint that the marginal distribution of predicted assignments for the last batches follows a prior long-tail distribution (Fig. 2A, *bottom*). With such a joint training objective, our network is encouraged to learn to distinguish a large variety of classes present in the regions sampled by RPN, while retaining the capability to classify *known* objects.

Secondary classification head architecture. As the box features $\hat{\mathbf{f}}$ are kept frozen during the discovery phase, we use several non-linear projection layers in the secondary classification head: $\hat{\mathbf{f}}^n = g^n(\hat{\mathbf{f}})$, where g^n is a multi-layer perceptron with ReLU activations. Such learned transformation is designed to better disentangle feature representation of novel classes without affecting the RoI box features that the known head is based on. On top of the projected features $\hat{\mathbf{f}}^n$ we employ a cosine classification layer, shown to be effective in the context of few-shot learning [20, 13, 67]: $\mathbf{l}^n = \frac{\mathbf{W}^n}{||\mathbf{W}^n||} \cdot \frac{\hat{\mathbf{f}}^n}{||\hat{\mathbf{f}}^n||}$, where $\mathbf{l}^n \in \mathbb{R}^N$ are the predicted novel-class logits. We determine the number of novel classes (*i.e.* the number of novel classes, N, in the secondary head) empirically in (Sec. 4.2).

Self-supervision via constrained clustering. We perform self-labeling via constrained clustering, where we assign a soft pseudo-label to each RPN proposal. We generate pseudo-labels $\mathbf{q}$ for all input images from $D^N \cup D^K$ that span all classes, both known and novel: $\mathbf{q} \in \mathbb{R}^{(N+K)\times B}$, where B is the batch size. We use $\mathbf{q}$ to compute the cross-entropy loss L_{ss} and train the classification heads. Standard clustering methods (*e.g.* k-means [47]) were shown to produce unstable results when employed for pseudo-labeling [9, 10]. Asano et al [2] instead propose to leverage an equipartitioning regularizer and [1] further extended the method to arbitrary prior distributions. Following [1], we aim to minimize a clustering energy function (*i.e.* cross-entropy loss): $E(\hat{\mathbf{p}}, \mathbf{q}) = \frac{1}{B}\sum_{i=1}^{B}\sum_{y}^{N+K} q(y|\mathbf{I}_i) \log \hat{p}(y|\mathbf{I}_i)$, where $\mathbf{q}$ are class labels (*i.e.* pseudo-labels in our framework), and $\hat{\mathbf{p}} = \text{softmax}([\mathbf{l}^k, \mathbf{l}^n]) \in \mathbb{R}^{(N+K)\times B}$ are predicted class-probabilities. Note that both labels $\mathbf{q}$ and probabilities $\hat{\mathbf{p}}$ are subject to optimization. Similar to Asano et al [2], we employ an alternating algorithm, where a constrained clustering method to generate $\mathbf{q}$ is followed by optimizing $\hat{\mathbf{p}}$, *i.e.*, during each training iteration, we first generate class pseudo-labels $\mathbf{q}$, and then use them to train the classification heads. To generate class pseudo-labels, we first choose a target clusters' marginal probability distribution (*i.e.* prior distribution). In our experiments, as prior, we use lognormal distribution. We then generate such soft pseudo-labels $\mathbf{q}$ that (i) closely match the chosen marginals and (ii) minimize $E(\hat{\mathbf{p}}, \mathbf{q})$ for the given $\hat{\mathbf{p}}$. In [2, 1], it is shown that this can be posed as a

constrained optimization problem and solved as an optimal transport problem using the fast online Sinkhorn-Knopp algorithm [14]. Having obtained the pseudo-labels $\mathbf{q}$, we fix them, calculate the loss $L_{ss} = E(\hat{\mathbf{p}}, \mathbf{q})$, and apply backpropagation to the classification heads' parameters.

Memory module. To generate good-quality pseudo-labels, we need a diverse set of samples in mini-batches in order to capture a wide variety of classes. With only a small set of features per mini-batch, we would likely produce noisy cluster assignments. As we operate in scenarios with more than a thousand classes to discover, we introduce a memory module (*c.f.*, [28, 11]) to store proposals' RoI box features from the last batches and ensure diverse set of samples when computing clustering assignments. As in [28, 11] we design the memory module as a queue, where at the end of each iteration we replace the old set of stored features with the features from the current batch. During pseudo-labeling generation we pass the extended set of features to the Sinkhorn algorithm. Specifically, after extracting current-batch proposals' features, we concatenate them with the features stored in the memory module $\mathbf{M}$, compute their logits, and as an input to the Sinkhorn algorithm use: $\hat{\mathbf{l}} = [\hat{\mathbf{l}}^k, \hat{\mathbf{l}}^n] = [h^k([\hat{\mathbf{f}}, \mathbf{M}]), h^n([\hat{\mathbf{f}}, \mathbf{M}])]$, where $\mathbf{M} \in \mathbb{R}^{M_{sz} \times \vec{F}}$ and M_{sz} is the number of features stored in the memory. Such modified procedure generates $B + M_{sz}$ pseudo-labels, for both current-batch proposals and memory features. We discard pseudo-labels generated for memory features and use only those B generated for the current-batch proposals, *i.e.*, we use the memory module only to improve the quality of current-batch pseudo-labels.

Swapping multi-view cluster assignments. We employ supervisory signal from multiple augmentations [11] to enforce the network to output consistent classification predictions $\hat{\mathbf{p}}$. Specifically, for each image from $D^N \cup D^K$, we first generate RPN proposals and refine their coordinates via class-agnostic classification head. We then regress the proposals' coordinates, and generate it's two augmented views (we discuss augmentations in the supplementary), and extract features for each. Then, we generate pseudo-labels for each set of features individually, obtaining two sets of pseudo-labels $\mathbf{q}_i$, $i \in \{1, 2\}$. Now, instead of computing the losses for each view as $E(\hat{\mathbf{p}}_i, \mathbf{q}_i)$, we swap the resulting pseudo-labels $\mathbf{q}_i$ between views, and calculate losses as $E(\hat{\mathbf{p}}_i, \mathbf{q}_j)$, where $i \neq j$, *i.e.* we use pseudo-labels obtained for one view, as labels to the other. We calculate the total loss as the average of the losses for both views: $\hat{L}_{ss} = (E(\hat{\mathbf{p}}_1, \mathbf{q}_2) + E(\hat{\mathbf{p}}_2, \mathbf{q}_1))/2$. For each view, we use a separate feature memory module.

Inference and evaluation. During inference (Fig. 2B), for each image, we extract a set of RPN proposals $\{(\mathbf{b}, o)\}$ and pass them through RoI heads for localization refinement and class predictions. For classification, we use both heads h^k, h^n and calculate probabilities as $\hat{\mathbf{p}} = \text{softmax}([\mathbf{l}^k, \mathbf{l}^n])$.

For quantitative evaluation, as our framework provides arbitrary class predictions (*i.e.* not directly mapped to novel classes in the target domain), we have to obtain a mapping of predicted classes with ground truth labels. We follow prior art [26, 25, 19, 68] and propose a similar evaluation methodology. First, we generate classification predictions for each GT annotation in the validation dataset. Then, we use the Hungarian algorithm [38] to obtain an one-to-one mapping between GT labels and validation instances based on these class predictions. Given this mapping, we ignore instances of predicted classes that were not mapped to GT classes. We proceed to the standard R-CNN inference pass, where we apply the foregoing mapping right after generating class predictions, and follow the standard R-CNN post-processing and evaluation [44].

4 Experimental Evaluation

In this section, we discuss our evaluation test-bed, including datasets, evaluation settings, and metrics (Sec. 4.1). In Sec. 4.2 we justify our main design decisions by studying NCDL performance in a well-controlled setting that closely follows a real-world scenario (training on a source *labeled* and target *unlabeled* dataset). We then compare our method's performance to several baselines (Sec. 4.3) and, finally, demonstrate that our method is applicable beyond our evaluation test-bed (Sec. 4.4).

4.1 Evaluation setting

COCO$_{half}$ + LVIS. We re-purpose COCO 2017 [44] and LVIS v1 [24] datasets for running ablations and comparisons with baselines and prior art. We use annotations for 80 COCO classes during the supervised training phase, aiming at further classifying additional 1000+ LVIS classes. COCO dataset contains $123K$ images with modal bounding box and segmentation mask annotations for 80 classes. The LVIS dataset contains a subset of $120K$ COCO images with annotations for 1203 classes

Table 1: Impact of supervised loss strength. We check the results of our method as function of strength of the supervised loss.

Sup. loss coeff.	mAP$_{all}$	mAP$_{known}$	mAP$_{novel}$
0	6.59	16.58	5.77
0.1	6.90	22.26	5.64
0.5	**6.92**	**25.00**	**5.42**
1.0	6.35	25.81	4.75

Table 2: Sensitivity to number of classes. We vary the number of novel classes set during the discovery phase.

# novel classes	mAP$_{all}$	mAP$_{known}$	mAP$_{novel}$
1000	5.42	23.24	3.95
3000	**6.92**	**25.00**	**5.42**
5000	6.24	24.91	4.70

that include all classes from COCO. For our experiments, we follow LVIS training and validation splits, resulting in $100K$ training images and $20K$ validation images. To model the NCDL setup, we further split $100K$ training images in half and use only $50K$ images with 80-class annotations from COCO during the supervised training phase. We refer to this dataset as COCO$_{half}$. We treat the rest of $50K$ images as additional unlabeled data used during the discovery phase.

LVIS + Visual Genome. We perform a large-scale generalization experiment by using annotations for 1203 LVIS classes during the supervised learning (*i.e.*, treating LVIS classes as *labeled*), aiming to learn to discover extra 2700+ classes from the Visual Genome (VG) v1.4 dataset [37] (*i.e.*, treating VG classes as *unlabeled*). We provide dataset details and splits (that ensure LVIS and VG class vocabularies do not overlap) in the supplementary. We note that annotations provided in VG are not exhaustive per class and many of its classes are abstract or semantically overlapping both within VG and when compared to LVIS. We thus mainly focus on qualitative results.

Implementation details. We base our model on Mask R-CNN [27] and FPN [43] implementations from Detectron2 [69]. We give all the implementation details in the supplementary material.

Evaluation metrics. For quantitative evaluation, we follow [26, 25, 19, 68] to match predicted cluster assignments with annotated semantic classes. We follow common practice and report mean average precision (mAP@[.5:.95]) [44]. For qualitative results, we include instances of classes that were matched to annotated classes only. We detail our evaluation procedure in the supplementary.

Baselines. We compare our end-to-end trainable detector with state-of-the-art Weng *et al.* [68] work, NCD methods ORCA [7], UNO[19], and k-means [47] baseline that operate on cropped object instances. For NCD methods we use labeled instance annotations from COCO$_{half}$ dataset as *labeled* images pool and cropped RPN proposals as *unlabeled* images pool. For proposals extraction we train R-CNN on COCO$_{half}$ under the same conditions as our base network, to ensure comparable evaluation conditions. Having trained the NCD classifiers, we use the same R-CNN to generate object proposals for the validation images, apply the classifiers on top of the proposals, and continue with the standard R-CNN post-processing steps. We provide more details on baselines in the supplementary.

4.2 Model ablations

We ablate single and multi-phase training, the impact of keeping supervised loss in the loop, the number of novel categories, pseudo-labeling prior, model architecture decisions, and backbone pretraining. In supplementary we provide further results on experiments with hyperparameters for proposals extractor and how class-agnostic heads affect the performance of the supervised baseline.

Single-phase training and unfreezing R-CNN components during discovery. As shown in Table 3 we observe that training the network with both supervised and discovery losses from the beginning, summed up without any scaling, leads to divergence. We hypothesize that this is due to noisy gradients from our self-supervised signal, with R-CNN being very sensitive to the choice of losses and hyperparameters [53]. To further support this claim, we experiment with a standard supervised phase followed by a discovery phase without freezing any of the components. We observe that such a setup also leads to divergence. Unfreezing only the RoI heads during the discovery phase, however, does not lead to divergence, yet reduces the score by 1.32 mAP. We thus keep all the components frozen during the discovery phase, apart from known and novel classification heads.

The strength of the supervised loss. We ablate the strength of the supervised classification loss that is used along with the discovery loss during the discovery phase. We vary the strength from 0 (no supervised loss) to 1 (no downscaling) and provide the results in Table 1. We observe that excluding supervised loss completely degrades the performance of the network for the known classes and benefits the performance of the novel classes. This indicates that there is a tradeoff between performance on known and novel classes, - keeping the full supervised loss introduces too much bias towards the former. We consequently set the strength of the supervised loss to 0.5 in all experiments.

Table 3: Additional ablations. We experiment with removing individual components from the network and their impact on the overall performance.

Method	mAP_{all}	mAP_{known}	mAP_{novel}
RNCDL	**6.92**	**25.00**	**5.42**
Single-phase framework	n/a	n/a	n/a
Unfreeze all layers	n/a	n/a	n/a
Unfreeze RoI heads	5.60	20.55	4.37
Uniform class prior	5.75	24.11	4.23
k-means [47] pseudo-lab.	2.06	21.13	0.48
W/o projection MLP	5.30	26.13	3.58
W/o swapped assignments	6.74	26.02	5.14
W/o memory	2.83	24.19	1.06
W/o MoCo pretraining	4.77	16.92	3.77

Table 4: Comparison with state-of-the-art models. * as per open source code. † adapted to support known classes in target dataset. ‡ randomly initialized novel head.

Method	mAP_{all}	mAP_{known}	mAP_{novel}
Methods that operate on cropped proposals			
k-means [47]	1.33	15.61	0.14
Weng *et al.* [68]*	1.62	17.85	0.27
ORCA [7]	2.03	20.57	0.49
UNO [19]‡	2.18	21.09	0.61
Methods that operate on FPN-based features			
k-means [47]	1.55	17.77	0.20
RNCDL w/ random init.‡	1.95	23.51	0.17
RNCDL	**6.92**	**25.00**	**5.42**
Fully-supervised	18.47	39.38	16.74

Sensitivity to the number of novel classes. In Table 2 we experiment with varying the number of novel classes we use for the novel classification head. We observe that the best number of classes for the COCO + LVIS setup is 3000, while other choices result in lower scores. We conjecture that such scores are sensitive to the semantic classes defined by the target dataset - with 3000 classes the model learns the granularity and grouping that matches closest to the LVIS classes, but that does not necessarily mean that a finer or coarser granularity categorization is worse. In supplementary we also experiment with attaching and training multiple heads jointly during the discovery phase and observe that scores slightly degrade in such setup but do not observe a significant drop in results.

Clustering prior. In Table 3 we experiment with using uniform prior distribution for class-marginals during pseudo-labels generation, as proposed in [2, 10]. We observe degradation of the score by 1.17 mAP, which confirms our hypothesis that a non-uniform lognormal prior distribution closer resembles a real-world long-tail class distribution. As an additional ablation, we experiment with using k-means for online pseudo-labels generation instead of constrained clustering and observe a significant degradation of the score by 4.86 mAP, especially notable for the novel classes.

Embedding projector and swapped assignments. In Table 3 we demonstrate the effect of adding an MLP feature projector on top of RoI box features for novel heads classification. We observe that the performance of the network improves by 1.62 mAP. In the same table, we show that swapping pseudo-labels between proposals under multiple augmentations helps boost the score by 0.18 mAP.

Feature memory module. We demonstrate the importance of memory module in Table 3. In all our experiments we used the memory size of 100 batches, unless stated otherwise. We observe that removing the memory module significantly drops the performance by 4.09 mAP, especially for the novel classes. This highlights that for an efficient pseudo-labels generation it is crucial to use a set of features representative of the full feature space that we lack in the absence of memory.

Self-supervised pre-training matters. We compare the results of our framework under different backbone initialization employed for the supervised pre-training stage. For the randomly initialized backbone we set all layers as trainable, while during the MoCo initialization the first two convolutional blocks are kept frozen (*c.f.*, [28]). In Table 3 we observe that MoCo-initialized backbone improves the score by 2.15 mAP over randomly initialized backbone mode.

Proposals extractor hyperparameters. We observe that when extracting the RPN proposals for self-supervised learning during the discovery phase the absence of NMS yields the best results and the optimal number of proposals per image is 50. We thus use such parameters for all the experiments.

4.3 Comparison to prior work

In Table 4 we compare our method with k-means baselines and three recent state-of-the-art methods that we adapted to our scenario. We present the results for both known (COCO) and unknown (the rest of LVIS) classes. Our method significantly outperforms the previous best NCD method UNO [19], by reaching 4.74 higher overall mAP. We outperform UNO in both the known and especially novel classes, where we reach 4.81 higher mAP. We also significantly outperform approach by Weng *et al.* [68].Our method is the only method to reach over 1 mAP in novel classes, where we reach 5.42 mAP. We believe the main advantages of our method to be the memory module, critical for effective self-supervision, and a backbone trained in an end-to-end manner with the aid of class-agnostic losses, further frozen during the discovery phase. Further, our backbone benefits from high-level semantic features via FPN that is critical for the classification of small objects [43]. We provide

a) COCO → LVIS b) LVIS → VisualGenome

Figure 3: Visualization of predictions for validation images of fully-supervised model and our RNCDL framework. We color the discovered novel classes in red.

a) COCO → LVIS b) LVIS → VisualGenome

Figure 4: Visualization of the common novel classes discovered in LVIS and VisualGenome datasets.

more comparison details in the supplementary. We note that as our method heavily relies on the generated region proposals, we are limited by the performance of region proposal network. It also does not update the backbone feature extractor, which could further boost the results, including the performance of localization and segmentation components.

4.4 Cross-dataset Generalization and Qualitative Results

We analyze the generalization properties of our framework by training on LVIS and evaluating on VG dataset. We present quantitative results in Table 5.

In Figure 3 we provide qualitative results with novel classes discovered. In Figure 4 we also visualize the common discovered categories. Upon manual examination of the clusters from the LVIS + VG experiment, we observe that many of the novel classes are synonyms, hyponyms, or hypernyms of known classes that appear in the LVIS dataset.

Table 5: LVIS → VisualGenome comparisons with a fully-supervised method.

Method	mAP_{all}	mAP_{known}	mAP_{novel}
Ours w/ random init.	2.13	7.71	0.81
Ours	4.46	12.55	2.56
Fully-supervised	4.52	13.72	2.35

5 Conclusions

In this paper, we introduce an end-to-end RNCDL network for the task of novel class discovery, detection, and localization. Our model is a two-stage object detection network that is able to classify both instances of the labeled, known classes and those of unlabeled, novel classes. At the core of our framework is a self-supervision guided by features of region proposals and a constraint for the novel class assignments to follow a long-tail distribution. In our experiments we demonstrate a significant improvement over the previous state-of-the-art. Furthermore, we demonstrate the ability to detect semantic classes without any supervision at a large scale on the Visual Genome dataset.

Broader Impact

The success of deep learning, to a large extent, can be attributed to high-quality labeled datasets. These datasets, for the most part, contain only a fixed number of categories (classes). Deep learning models trained on such datasets can only make predictions for objects that belong to one of the training set categories. However, the world consists of an extremely large number of categories, and enlarging the training set with more labeled data for all the categories is an infeasible problem. Clearly, there is a need of shit towards solutions that are not limited by labeled data supervision. Novel class category discovery has been mostly tackled in image classification setup, but there have been no works that give suitable solutions to the more challenging task of object detection. In this work, we improve compared to previous methods by an order of magnitude.While the problem is still far from solved, we hope that our method makes the first concrete step toward an effective solution, and would inspire others to tackle this challenging problem. We expect that this method will lead to more object detection works that aim to discover instances of novel classes.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020.

[2] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018.

[4] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015.

[5] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, 2016.

[6] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[7] Kaidi Cao, Maria Brbić, and Jure Leskovec. Open-world semi-supervised learning. In *ICLR*, 2022.

[8] Yuhang Cao, Jiaqi Wang, Ying Jin, Tong Wu, Kai Chen, Ziwei Liu, and Dahua Lin. Few-shot object detection via association and discrimination. 2021.

[9] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[10] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[11] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[12] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, 2017.

[13] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICML*, 2019.

[14] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 2013.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[16] Akshay Raj Dhamija, Manuel Günther, and Terrance E Boult. Reducing network agnostophobia. In *NeurIPS*, 2018.

[17] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[18] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *CVPR*, 2021.

[19] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *ICCV*, 2021.

[20] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018.

[21] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[22] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[23] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[24] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019.

[25] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *ICLR*, 2020.

[26] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*, 2019.

[27] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[29] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Deep image category discovery using a transferred similarity function. *arXiv preprint arXiv:1612.01253*, 2016.

[30] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018.

[31] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to Segment Every Thing. In *CVPR*, 2018.

[32] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *CVPR*, 2021.

[33] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, 2014.

[34] Xuhui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single-and multi-modal data. In *ICCV*, 2021.

[35] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021.

[36] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019.

[37] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[38] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[39] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *IJCV*, 128(7):1956–1981, 2020.

[40] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised object discovery and tracking in video collections. In *ICCV*, 2015.

[41] Yong Jae Lee and Kristen Grauman. Object-graphs for context-aware visual category discovery. In *CVPR*, 2010.

[42] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, 2011.

[43] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[44] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[45] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 128(2):261–318, 2020.

[46] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[47] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob., Vol. 1*, 1967.

[48] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *ICRA*, 2018.

[49] Aljoša Ošep, Paul Voigtlaender, Jonathon Luiten, Stefan Breuers, and Bastian Leibe. Large-scale object mining for object discovery from unlabeled video. In *ICRA*, 2019.

[50] Aljoša Ošep, Paul Voigtlaender, Mark Weber, Jonathon Luiten, and Bastian Leibe. 4d generic video object proposals. In *ICRA*, 2020.

[51] Shafin Rahman, Salman Hameed Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. *ACCV*, 2018.

[52] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017.

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[54] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[55] Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[56] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *PAMI*, 35(7):1757–1772, 2012.

[57] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *PAMI*, 36(11):2317–2324, 2014.

[58] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[59] Josef Sivic, Bryan C. Russell, Andrew Zisserman, William T. Freeman, and Alexei A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.

[60] Tinne Tuytelaars, Christoph H. Lampert, Matthew B. Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. *IJCV*, 88:284–302, 2010.

[61] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *ECCV*, 2020.

[62] Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019.

[63] Huy V Vo, Patrick Pérez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *ECCV*, 2020.

[64] Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *NeurIPS*, 2021.

[65] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[66] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.

[67] Xin Wang, Thomas E. Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020.

[68] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *CVPR*, 2021.

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

[70] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *PAMI*, 2018.

[71] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[72] Bingchen Zhao and Kai Han. Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. In *NeurIPS*, 2021.

[73] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, 2021.

[74] Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *CVPR*, 2021.

[75] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection. In *CVPR*, 2022.