To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

Learning the Easy Things First: Self-Paced Visual Category Discovery

Yong Jae Lee and Kristen Grauman
University of Texas at Austin
yjlee0222@mail.utexas.edu, grauman@cs.utexas.edu

Abstract

*Objects vary in their visual complexity, yet existing discovery methods perform "batch" clustering, paying equal attention to **all** instances simultaneously—regardless of the strength of their appearance or context cues. We propose a self-paced approach that instead focuses on the **easiest** instances first, and progressively expands its repertoire to include more complex objects. Easier regions are defined as those with both high likelihood of generic objectness and high familiarity of surrounding objects. At each cycle of the discovery process, we re-estimate the easiness of each subwindow in the pool of unlabeled images, and then retrieve a single prominent cluster from among the easiest instances. Critically, as the system gradually accumulates models, each new (more difficult) discovery benefits from the context provided by earlier discoveries. Our experiments demonstrate the clear advantages of self-paced discovery relative to conventional batch approaches, including both more accurate summarization as well as stronger predictive models for novel data.*

1. Introduction

Visual category discovery is the problem of extracting compact, object-level models from a pool of unlabeled image data. It has a number of useful applications, including (1) automatically summarizing the key visual concepts in large unstructured image and video collections, (2) reducing human annotation effort when constructing labeled datasets to train supervised learning algorithms, and (3) detecting novel or unusual patterns that appear over time.

Problem. Existing methods treat unsupervised category discovery as a one-pass "batch" procedure: the input is a set of unlabeled images, and the output is a set of k discovered categories found via clustering or topic models [19, 13, 6, 11, 23]. Such an approach implicitly assumes that all categories are of similar complexity, and that all information relevant to learning is available at once. However, paying equal attention to *all* instances makes the grouping sensitive to outliers, and can skew the resulting models unpredictably. Furthermore, it denies the possibility of exploiting inter-object context cues during discovery; one cannot detect the typical relationships between objects if models for the component objects are themselves not yet formed.

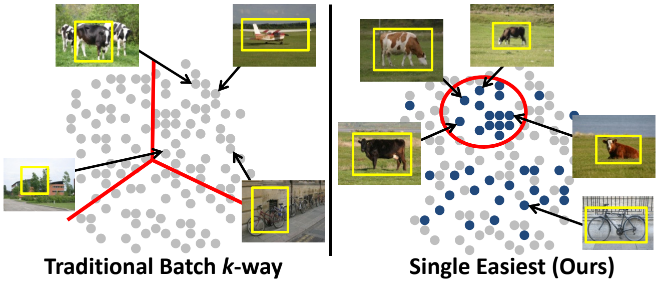


Figure 1. In contrast to traditional k-way batch clustering approaches (left), we propose to discover "easier" objects first. At each cycle of discovery, a measure of easiness isolates instances more amenable to grouping (darker dots on right).

Idea. Instead, we propose a self-paced approach to visual discovery. The goal is to focus on the "easier" instances first, and gradually discover new models of increasing complexity. What makes some image regions easier than others? And why should it matter in what order objects are discovered? Intuitively, regions spanning a single object exhibit more regularity in their appearance than those spanning multiple objects or parts thereof, making them more apparent for a clustering algorithm to group. At the same time, regions surrounded by familiar objects have stronger context that can also make a grouping more apparent. For example, if the system discovers models for desks and computer monitors first, it is then better equipped to discover keyboards in their midst. In contrast, if it can currently only recognize kitchen objects, keyboards are less likely to emerge as an apparent cluster.

In human learning, it is common that easier concepts help shape the understanding of more difficult (but related) concepts. In math, one learns addition before multiplication; in CS, linked lists before binary trees. We aim to capture a similar strategy for visual discovery. However, a critical distinction is that our approach must accumulate its discoveries without any such prescribed curriculum. That is, it must self-select which aspects to discover first.

Approach overview. To implement this idea, we first introduce a measure of easiness that uses two criteria *automatically* estimated from the unlabeled data: (1) the likelihood that the region represents a single object from any generic category—its "objectness"; and (2) the likelihood that its surrounding image regions are instances of familiar categories for which we have trained models (i.e., the familiarity of the surrounding regions)—its "context-awareness".

Initially the system is equipped with models of "stuff" categories (grass, sky, etc.). Then, given an unlabeled image collection, it proceeds to discover "things" (objects) a *single* category at a time, in order of predicted easiness. After each discovery, we update the set of familiar categories by training a detector for the newly found object class, which allows us to produce a richer context model for each remaining (harder) unfamiliar instance. Similarly, we revise the easiness estimates on all data, and loosen the easiness criterion for the next round of discovery. Thus, in contrast to a one-pass k-way partitioning, our approach gradually accumulates models for larger portions of the data. The process continues until all data is either accounted for, or else fails to meet the least selective easiness criterion. See Fig. 1.

Our main contribution is the idea of visual discovery through a self-paced curriculum. We validate all aspects of our approach on realistic natural images, and show clear advantages for summarization compared to conventional batch clustering and state-of-the-art discovery algorithms. Further, we show that we can train models to predict instances in novel images in an interactive setting where a human annotator names each discovered category. We achieve competitive results to fully supervised baselines at a fraction of the required human labeling cost.

2. Related Work

Unsupervised visual discovery methods detect appearance patterns using either clustering algorithms [10, 6, 11] or latent topic models [19, 13]. Our previous approach [11] shows how familiar objects surrounding a region of interest can help identify more accurate clusters. However, it attempts to discover all categories at once, and restricts the context to pre-specified familiar categories learned from labeled data. Whereas all existing discovery methods adhere to the traditional batch framework, we propose to learn easier categories first and incrementally expand the context with each discovery.

While most supervised object recognition methods also take a one-pass learning approach, a few researchers consider ways to progressively enhance a model. A dataset collection technique in [12] is initialized with labeled seed images, and then incrementally expanded with keyword search. Bootstrap learning methods iteratively aggregate results from learners of increasing complexity, and have been explored in robotics [7]. Recent work in deep feature hierarchies use unsupervised learning to build mid-level features that better serve some supervised prediction task (e.g., [18]). Our motivation to expand an intermediate representation via sequential discoveries is related, though we work with unlabeled data.

Various forms of context have been explored in the object recognition literature; several methods model the co-occurrence and relative spatial layout of objects to improve predictions (e.g., [20, 5, 15, 22, 9]). The method of [5] uses automatically discovered "stuff" context to improve detection of "things", given labeled data for each object of interest. Our initial context model is derived from "stuff", but otherwise our goal to discover objects in unlabeled data is quite different.

Researchers have proposed approaches to generate category-independent hypotheses from image regions. Unique pixels that stand out from their surroundings can be used to detect salient objects [14]. The "objectness" measure [1] is used to bias a sampling procedure for potential object bounding boxes, while the method of [4] generates a ranking over candidate regions. Whereas these methods aim to improve detection speed, we explore how objectness can help identify easier regions for discovery.

Recent work explores how a curriculum that sorts training data into easier and harder groups can lead to better local minima when training a classifier with a non-convex criterion [3]. Our idea is similarly motivated. However, in contrast to our approach, the strong assumption in [3] is that the curriculum can be provided by a human teacher (e.g., when learning shapes, it is shown special cases like squares and circles before being exposed to rectangles and ellipses). A curriculum-based approach for structural SVM training shows how to simultaneously estimate model parameters and easiest exemplars [8], but its determination of easiness is tied to the training objective on labeled data. In contrast to both approaches, we have no such top-down curriculum in the unsupervised setting, and instead show how the system itself can predict the easiest instances.

3. Approach

Our goal is to discover visual categories from an unlabeled image collection by grouping image regions with similar appearance and context.[1] Throughout the discovery process, we maintain two disjoint sets of image subwindows: $\mathcal{D}$, the discovered windows that have been assigned to a cluster, and $\mathcal{U}$, the undiscovered windows that remain in the general unlabeled pool. In addition, we maintain an evolving set of familiar categories $\mathcal{C}_t = \{c_1, \dots, c_{N_t}\}$, where N_t is the category count at iteration t. Initially $\mathcal{D}$ is empty.

Our approach iterates over four main steps: (1) iden-

[1] We use "region", "subwindow", and "window" interchangeably.

tifying the easy instances among the image regions in $\mathcal{U}$; (2) discovering the next prominent group of easy regions; (3) training a model with the discovered category to detect harder instances in the data, moving them to $\mathcal{D}$; and (4) revising the object-level context for all regions in $\mathcal{U}$ according to the most recent discovery. We first explain how we represent a cluster (Sec. 3.1), and how we initialize the set of familiar categories (Sec. 3.2). We then describe each of the four main steps in turn (Secs. 3.3 to 3.6).

3.1. Exemplar-based Category Models

We use a simple exemplar-based model to represent familiar classes, i.e., those the system has discovered thus far. Each region or window is represented by T types of texture/color descriptors (to be defined in Sec. 4). The likelihood of region $r \in \mathcal{U}$ given class $c_j \in \mathcal{C}_t$ is defined by its mean affinity to all instances that were grouped together to form class c_j:

$$P(r|c_j) \propto \frac{1}{T}\sum_{m=1}^{T}\frac{1}{|c_j|}\sum_{l\in c_j} K_m(r,l), \tag{1}$$

for $j = 1, \ldots, N_t$, where l indexes the exemplars in category j, and each K_m is a χ^2 kernel computed on the m-th feature type. These likelihood values are used below to capture how familiar regions appear to be.

3.2. Initializing the Pool of Familiar Categories

We initialize the familiar set $\mathcal{C}_0$ with classifiers for "stuff" categories, which are materials with regular fine-scale features, but no specific spatial shape, e.g., *grass, sky, water, road, leaves*. Stuff classes can be classified quite accurately, and are typically widespread in natural scenes. We therefore choose to use them as initial context, and allow the approach to immediately focus on discovering "things"—categories with well-defined shape that often appear amongst the stuff. Thus we populate $\{c_1, \ldots, c_{N_0}\}$ with true instances of the N_0 stuff classes. Given a novel image in the unlabeled collection, we generate its bottom-up segmentation (we use [2]), and can compute each region's likelihoods as defined in Eqn. 1.

3.3. Identifying Easy Objects

Next we proceed to identify the easiest instances among $\mathcal{U}$ according to both low-level image properties and the current familiar classes in $\mathcal{C}_t$. We define an "easiness" function

$$ES(w, \mathcal{C}_t) = Obj(w) + CA(w, \mathcal{C}_t) \tag{2}$$

that scores a window w based on how likely it is to contain an object ("objectness", Obj) and to what extent it is surrounded by *familiar* objects ("context-awareness", CA).

We compute "objectness" to capture how well an image region appears to contain an object of *any* generic class.

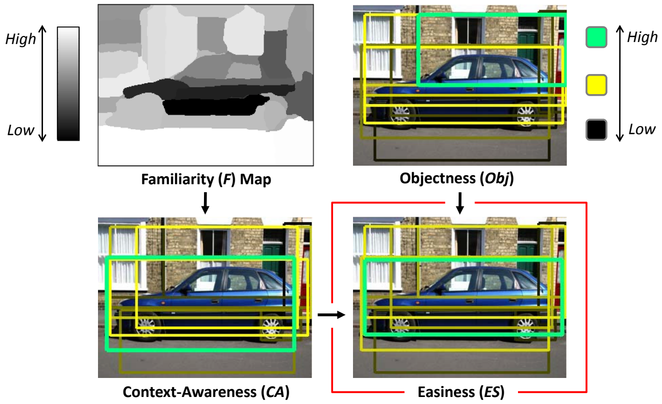


Figure 2. Objectness and context-awareness both influence the "Easiness" estimates. Context-awareness favors subwindows surrounded by familiar things, while objectness favors windows surrounding a thing appearing well-separated from background.

Note, this measure does *not* care about class familiarity, it reflects only the generic object-like properties of the window (saliency, apparent separation from background, etc.) We generate candidate regions using the measure developed in [1]. It uses a Bayesian classifier based on multiscale-saliency, color contrast, and superpixel straddling cues to compute the probability that a window contains any object, and is trained using unrelated image data.[2] For each image, we sample 10,000 windows uniformly across the image at multiple scales, and compute the objectness score $Obj(w)$ for each window. We then sample 50 windows according to the resulting objectness distribution (see Fig. 2, top right).

We compute "context-awareness" to capture how closely an image window's surrounding regions resemble familiar categories. We first compute the likelihoods defined in Sec. 3.1 for each image region; we average the values at any pixels covered by multiple partially overlapping regions. Using those probabilities, we compute a superpixel *familiarity map*, where the familiarity of superpixel s is:

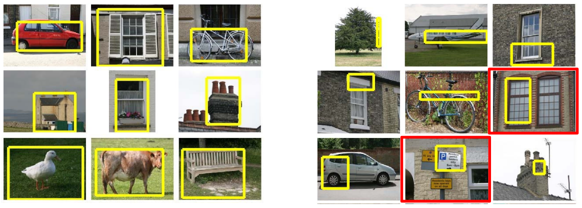
$$F(s, \mathcal{C}_t) = \max_{c_j \in \mathcal{C}_t} P(s|c_j), \tag{3}$$

where the $\max$ reflects we care only about the degree to which s belongs to *any* familiar category. (See Fig. 2, top left).[3]

Let $s_1(w), \ldots s_R(w)$ denote the R spatially nearest superpixels surrounding window w, in order of proximity. The final context-awareness score is a spatially weighted aver-

[2] We use the authors' code, which was built with INRIA Person, Pascal 06, and Caltech 101 images [1].

[3] The role of the superpixels is simply to summarize measurements coherently within local regions in the image, and ensure we cover regular regions around each window; however, note that the original likelihoods were computed from regions with larger spatial extents.



(a) Easy Instances (b) Hard Instances

Figure 3. Randomly selected examples among the easiest and hardest instances chosen by our algorithm. Our method is able to bypass the hard or noisy instances, and focus on the easiest ones first. Note that a region with high objectness can yield low easiness if its context is yet unfamiliar (e.g., see red boxes in (b)).

age of their familiarity scores:

$$CA(w, \mathcal{C}_t) = \sum_{j=1}^{R} \mathrm{w}_j F\big(s_j(w), \mathcal{C}_t\big) \tag{4}$$

where $\mathrm{w}_j = R - j + 1$ serves to give regions nearest to the window the most influence. Before combining the component $Obj(\cdot)$ and $CA(\cdot)$ terms, we rescale by mapping their distributions to standard Gaussians.

We sort all unclustered instances in decreasing order of easiness (Eqn. 2); see Fig. 3 for examples. Then, we perform discovery on only the easiest instances, as determined by a threshold computed from the data: $\theta_t = 2\sigma - 0.1t$, where σ denotes the standard deviation of all easiness scores in $\mathcal{U}$ and t is the iteration of discovery. Since $ES(\cdot)$ has a standard Gaussian distribution, larger portions of its right tail are considered to be "easy" over the iterations.

3.4. Single Prominent Category Discovery

Thus far we have a way to model familiar discovered objects and to identify the easiest instances. Now we overview how we represent each easy instance, and then how we extract a single prominent cluster among them.

Representation for each instance: Given a candidate easy window $w \in \mathcal{U}$ at iteration t, we form an appearance $A(w)$ and context $G_t(w)$ descriptor. We use standard descriptors for appearance (e.g., pHOG; see Sec. 4), and a variant of the object-graph [11] for context. The object-graph pools the familiar category likelihoods for the window's spatially nearest superpixels, recording the values according to their relative layout. The resulting descriptor is a series of histograms:

$$G_t(w) = [H_1(w), \ldots, H_R(w)], \tag{5}$$

where for $i = 1, \ldots, R$ each component histogram

$$\begin{aligned} H_i(w) \quad = \quad & [\sum_{j=1}^{i} P(s_{j_a}(w)|\, c_1), \ldots, \sum_{j=1}^{i} P(s_{j_a}(w)|\, c_{N_t}) \\ & \sum_{j=1}^{i} P(s_{j_b}(w)|\, c_1), \ldots, \sum_{j=1}^{i} P(s_{j_b}(w)|\, c_{N_t})]. \end{aligned}$$

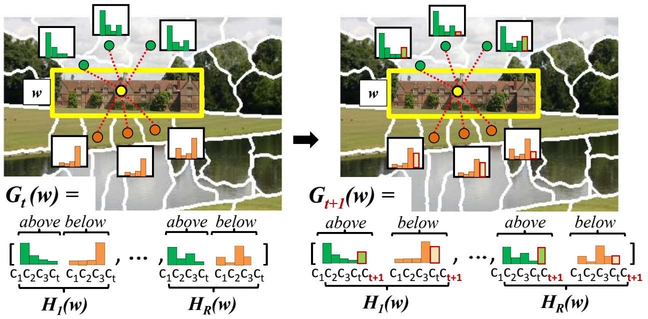


Figure 4. **(left)** The object-graph descriptor [11] for window w at iteration t. Each histogram $H_i(w)$ accumulates the likelihoods for the N_t familiar classes $(c_1, \ldots, c_t)$ over the nearest i superpixels, up to $i = R$. **(right)** The descriptor at iteration $t+1$. Note how it has expanded to reflect the most recent discovered category: c_{t+1}.

accumulates the likelihoods for the N_t familiar classes over the nearest i superpixels, where $s_{j_a}(w)$ denotes the j-th nearest superpixel above the window w, and $s_{j_b}(w)$ denotes the j-th nearest one below it. Nearness is determined based on region centroids. (See Fig. 4.)

To compute the similarity between two windows w_i and w_j, we use the combined kernel:

$$K(w_i, w_j) = K_{\chi^2}\left(A(w_i), A(w_j)\right) + K_{\chi^2}\left(G_t(w_i), G_t(w_j)\right), \tag{6}$$

where K_{χ^2} denotes a χ^2 kernel. Under this kernel, easy instances with both similar appearance and context are most likely to be grouped together.

Prominent category discovery: Given the current easy windows and the combined kernel, at each iteration we want to expand the pool of discovered categories with a single prominent cluster. Recall, the easiest instances already serve to focus the algorithm on those regions with consistent representations. In particular, our context-awareness criterion is directly linked to the data representation during clustering: the easiest instances are surrounded by familiar regions with relatively high likelihoods (see Eqns. 3 and 4), which makes comparisons between their object-graphs meaningful. Thus, by seeking a single new cluster, we can conservatively identify the most obvious new group; further, we can incrementally refine the context model most quickly for future discoveries.

To discover the most prominent category, we first partition the data into candidate groups, and then refine the most distinctive one. Specifically, we perform complete-link agglomerative clustering over the easy instances using the kernel in Eqn. 6, which offers robustness to outliers (i.e., windows that are poorly localized or contain rare objects) and allows us to target a cluster size rather than a cluster number. We stop merging when the distance between the most similar (yet-unclustered) instances becomes too large—specifically, greater than one standard deviation beyond the mean distance between all instances—and au-

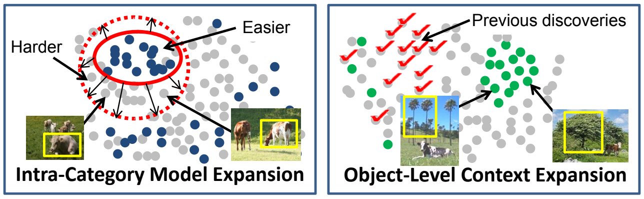

Harder
Easier
Intra-Category Model Expansion
Previous discoveries
Object-Level Context Expansion

Figure 5. Discovered category knowledge expansion. See Sec 3.5.

tomatically select the tightest cluster with the highest silhouette coefficient [21] among the candidate groups. We then refine the selected instances with Single-Cluster Spectral Graph Partitioning (SCSGP) [17, 16], which maximizes the average consensus. This step reduces possible outliers in the discovered group from agglomerative clustering.

We found this procedure to perform much better in practice than simply directly applying a "single-cluster" algorithm (e.g., Min Cut or SCSGP alone). This is likely due to the latter's sensitivity to a small number of outlying points, and the presence of overlapping clusters.

3.5. Discovered Category Knowledge Expansion

Each newfound discovery—a single prominent cluster identified among the easiest instances—serves to benefit later discoveries; this is a key property of our self-paced curriculum learning approach. In particular, it helps at both the *intra-category* and *inter-category* levels, as we explain next.

Intra-category model expansion: First, the initially discovered easier instances yield a model that can be used to detect the harder instances, which would not have clustered well due to their appearance or different context. We use instances in the newly discovered category to train a one-class SVM based on their appearance representation (no context). Then, we apply the classifier to all remaining windows in $\mathcal{U}$, merge the positively classified instances with the discovered category, and move them to $\mathcal{D}$.

While object-level context helps the discovery algorithm group the *easier* instances, we intentionally exclude context from the classifier's feature space for this stage. The goal is to be more inclusive and identify the harder instances of the class. For example, we might first discover cows in grass as the easy case, and then use the corresponding cow model to find other more challenging instances of cows that are partially occluded or surrounded by other animals (see Fig. 5, left; darker dots denote easier instances).

Object-level context expansion: Second, the expansion of the context model based on the discovered categories can help to discover certain harder ones. With each discovery, $\mathcal{C}_t$ expands. Thus, for every window remaining in $\mathcal{U}$, we revise its object-graph $G_t(\cdot)$ to form $G_{t+1}(\cdot)$, augmenting it with class affinities for the discovered category, per spatial component (see Fig. 4). This enriches the object-level context, altering both the feature space and the easiness scores. In effect, while we have weaker context models when detecting the easiest objects, we have richer context models when considering harder instances at later iterations. For example, having detected the "stuff" regions (grass, roads, sky), the system may discover cows in the simple meadow scenes, and then exploit its expanded context to later discover diverse-looking trees that appear in the context of both grass and cows (see Fig. 5, right).

```
Input: Unlabeled images; stuff models $c_1, \ldots, c_{N_0}$.
Initialize $\mathcal{U}$ with all regions from unlabeled inputs; $\mathcal{D} = \emptyset$;
$\mathcal{C}_0 = \{c_1, \ldots, c_{N_0}\}$; $t \leftarrow 1$.
while Easy instances remain in $\mathcal{U}$: do
    1. Identify easy instances $ES(w, \mathcal{C}_t) > \theta_t$ in $\mathcal{U}$.
    (Sec. 3.3)
    2. Discover single prominent category among them.
    (Sec. 3.4)
    3. Detect harder intra-class instances with one-class
    classifier; move instances to $\mathcal{D}$, add new category to $\mathcal{C}_t$.
    (Sec. 3.5)
    4. Expand context descriptor for each instance in $\mathcal{U}$.
    (Sec. 3.5)
    5. Revise familiarity map; recompute easiness.
    (Sec. 3.3)
    6. Loosen easiness criterion; $\theta_t = 2\sigma - 0.1t$. (Sec. 3.6)
    $t \leftarrow t + 1$
end
Output: Set of $t$ discovered categories in $\mathcal{D}$.
```

Algorithm 1: Algorithm recap

We validate the impact of both the intra-category model expansion and object-level context expansion on category discovery in Sec. 4, Figs. 6 and 9, respectively.

3.6. Iterative Discovery Loop

Finally, having augmented $\mathcal{C}_t$ with the newly discovered category, we proceed to discover the next easiest category. Note that the easiness scores evolve at each iteration of the discovery loop as more objects become familiar. Further, the annealing of the threshold defined in Sec. 3.3 essentially loosens the "easiest" criterion over time, allowing the algorithm to discover harder categories in later iterations, when context models are potentially richer. As the method iterates, it accounts for more instances.

We iterate the process until the remaining instances in $\mathcal{U}$ are too hard: this makes the system robust to noisy and rare instances that do not belong to any cluster. Alg. 1 summarizes the steps of our algorithm.

4. Results

Our experiments quantify our method's clustering and segmentation accuracy using standard metrics from previous work [19, 10, 6, 11], and we additionally demonstrate classification performance on novel images using models learned with the discovered categories.

Baselines: We compare to several baselines: 1) a side-by-side implementation of batch clustering, 2) a baseline that focuses on the hardest instances first (those with lowest easiness) but otherwise follows our pipeline, and 3) two existing state-of-the-art discovery methods [19, 11].

Dataset: We use the MSRC-v0 dataset, which consists of 3,457 natural scenes with 21 object classes (*building, tree, cow, sheep, car, bicycle, sign, window, grass, sky, mountain, airplane, water, flower, bird, chair, road, body, leaf, chimney, door*), and was previously studied in [19, 11]. The wide variety of categories allows us to properly evaluate the impact of both easiness selection and context refinement. We learn stuff classes on 40% of the data, and run discovery on the other 60%.[4] With 50 sampled windows per image, this makes 60,000 instances in the unlabeled pool.

Implementation details: We use [2] to obtain candidate stuff regions. We combine texture, color, and shape features to form $A(w)$ for window w. To describe texture, we compute SIFT bag-of-words histograms for the regions and Spatial Pyramid histograms for the windows; we densely sample 8-pixel wide SIFT patches at every pixel. To describe color, we use Lab color space histograms, with 23 bins per channel. To describe shape, we compute pHOG descriptors with 3 levels and 8 bins. For the object-graphs, we generate an over-segmentation with roughly 50 superpixels per image, and fix $R = 20$, following [11]. We normalize all histograms to sum to 1. We set $\nu = 0.1$ for the one-class SVM.

Evaluation metrics: To quantify discovery accuracy, we use *purity* [21], which is the percentage of correctly labeled instances, where all instances in a cluster are assigned to its majority class's true label. To score a window, we take its true label to be that to which the majority of its pixels belong. To quantify the segmentation accuracy of a window w, we use the pixel-level *overlap score*, $OS = \frac{|GT \cap w|}{|GT \cup w|}$, where GT is the ground-truth object segmentation, i.e., the tightest bounding box covering the full object region associated with w's majority pixel label.

Object discovery accuracy: We first analyze the quality of our discovered clusters, compared to both the batch and "hardest first" baselines. All methods use the same features and agglomerative clustering algorithm. The batch baseline is meant to show the limitations of existing methods, all of which determine k models in one pass over all the data. To ensure the batch baseline is competitive, we give it the non-overlapping windows with the highest objectness score per image as input.

Fig. 6 shows the results. We plot purity as a function of the *% of ground-truth object instances discovered* in order to analyze the quality of the discovered groups *and* quantify the recall rate for the true objects found. We count true objects as windows with at least 50% overlap with ground truth; if multiple windows overlap a ground-truth object, we score only one of them. Each point shows the result for a given number of clusters, for $k = t = [1, 40]$. At each iteration, our method finds about 5-15% of the instances to be "easy".

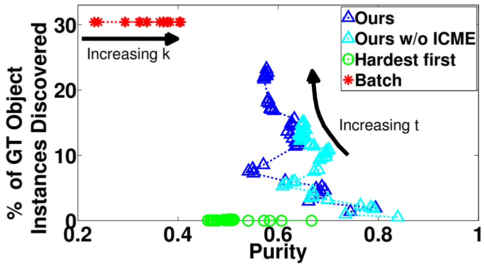


Figure 6. Discovery accuracy as a function of the percentage of unique object instances discovered. Our approach produces significantly more accurate clusters than either baseline, while selectively ignoring instances that cannot be grouped well.

Our approach provides significantly more accurate discoveries than either baseline. Note that purity increases with k for the batch method, since the k-way clusters computed over all windows get smaller, which by definition generates higher purity. In contrast, our method accounts for *more* windows as t increases, and purity gradually declines as the easiness criterion is relaxed. This difference highlights the core concept behind our approach: rather than force k splits, it steadily and selectively increases its pool of discovered objects. It purposely does not integrate all possible object instances (ignoring harder or poorly grouped ones), and yields accuracy more than twice as good as the batch approach. (In Table 1, we show the impact that this has on generalization performance.) For reference, the upper bound on instances we could discover is 53%, which is the portion of true objects present in the initial 50 windows per image. Most of the missed objects (for any method) are small object parts, e.g., windows or doors on cars, or objects that are not well-represented with windows, e.g., walls that are labeled as "building" in the ground truth.

Our substantial improvement over the "hardest-first" baseline validates our claim that considering the easiest instances per iteration leads to more accurate models. It also indicates that the easiest instances are indeed those that best capture true object regions. Note that while the hardest-first baseline technically has higher purity than batch, it discovers almost no objects—most windows it chooses to group overlap multiple objects or object parts.

Finally, the plot also reveals the impact of our intra-category model expansion. By using models discovered on easier examples to classify harder instances of the same ob-

[4] In all experiments we treat "stuff" classes as initial context, as explained in Sec. 3.2. While in principle one could also use our framework with "things" as initial known classes, the implementation is not straightforward with the cues we chose (regions for stuff, windows for things). See [11] for results analyzing the impact of which classes serve as initial context for discovery.

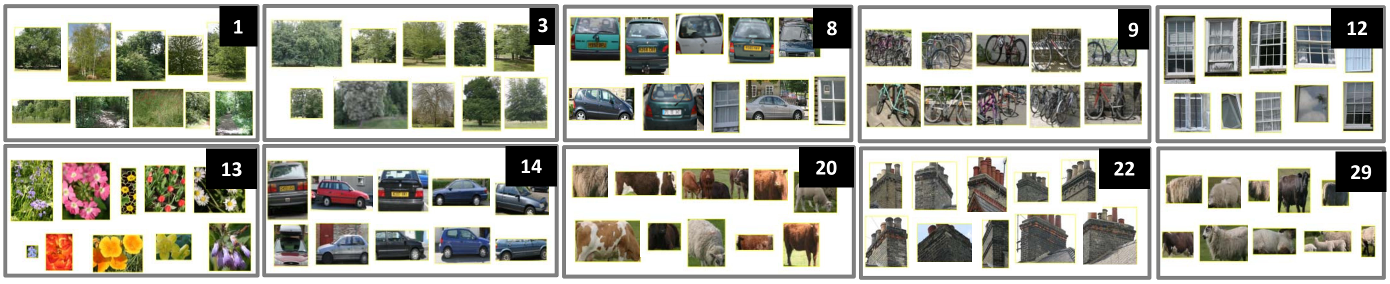


Figure 7. Examples of discovered categories; numbers indicate the iteration when that discovery was made. See text for details.

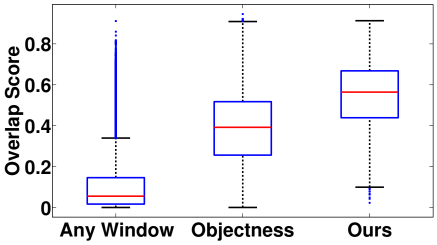


Figure 8. Object segmentation accuracy for random image windows (left), windows sampled by objectness alone (center), and those discovered by our approach (right). Higher values are better.

ject, we successfully discover a larger percentage of the instances in the data, with only a slight reduction in purity. (Compare "Ours" to "Ours w/o ICME" in Fig. 6.)

Fig. 7 shows representative example discoveries, sorted by iteration. We display the top 10 regions for each category, as determined by their silhouette scores. Note that the easiest categories (trees and bicycles) have high objectness and context-awareness scores, as well as strong texture, color, and context consistency, causing them to be discovered early on. The harder chimney and sheep objects are not discovered until later. There are some failure cases as well (see $t = 3, 8$), such as re-discovering a familiar category (trees) or merging different categories due to similar appearance (cars and windows).

Object segmentation accuracy: Since the images contain multiple objects, our algorithm must properly segment each object in order to obtain clusters that agree with semantic categories. Thus, we next compare the overlap accuracy for the object instances we discover in 40 categories to (1) the initial 50 windows sampled per image according to their objectness scores, and (2) 50 *randomly* sampled windows per image.

Fig. 8 shows the results. The windows sampled according to objectness are already significantly better than the random baseline, showing the contribution we get from the method of [1]. However, our method produces even stronger segmentations, showing the impact of the proposed context-awareness and easiness scoring.

Impact of expanding models of object context: Next we evaluate the impact of object-level context expansion. To isolate this aspect, we compare against a baseline that follows the same pipeline as our method, but uses familiar models for only the initial stuff categories; it does not update its context model after each discovery.

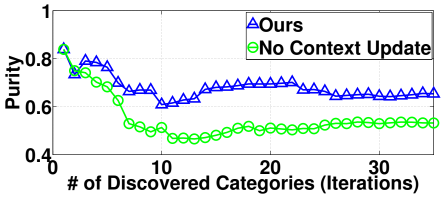


Figure 9. Impact of expanding the object-level context.

Fig. 9 shows the results, in terms of purity as a function of the number of discovered categories. As expected, the cluster quality is similar in the first few iterations, but then quickly degrades for the baseline. The first few discoveries consist of easy categories with familiar "stuff" surrounding them, and so the baseline performs similarly to our method. However, without any updates to the context model, it cannot accurately group the harder instances (e.g., cars, buildings). In contrast, by revising the object-level context with new discoveries, we obtain better results.

Comparison to state-of-the-art methods: We next compare against two existing state-of-the-art batch discovery algorithms: our object-graph method [11] and the Latent Dirichlet Allocation topic model method of Russell et al. [19]. These are the most relevant methods in the literature, since both perform discovery on images with multiple objects (other techniques generally assume a single object per image). We run all methods on the same MSRC data, and use publicly available source code, which includes feature extraction. To quantify how well each method summarizes the same data, we use the F-measure: $\frac{2 \cdot P \cdot R}{P+R}$, where P denotes precision and R denotes recall.[5] Since we do not know the optimal k value for any method, we generate results for a range of values and show the distribution (we consider $k = [10, 40]$, since the data contains 21 total objects). Fig. 10 shows that our method produces the most

[5] We evaluate recall with respect to each method's output discoveries, since the target categories are slightly different. The object-graph method and ours attempt to discover only the "things", while the topic model method attempts to discover all categories.

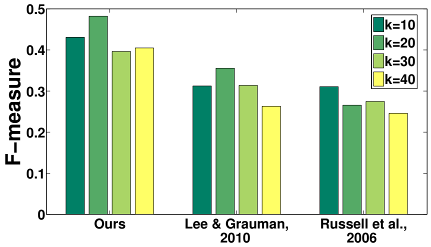


Figure 10. Comparison to state-of-the-art discovery methods. Our method summarizes the data more accurately than either baseline.

reliable summary of the unlabeled image data.

Predicting instances in novel images: Finally, we test whether the discovered categories generalize to novel images outside of the discovery pool. The goal is to test how well the system can reduce human effort in preparing data for supervised classifier construction. The discovery system presents its clusters to a human annotator for labels, then uses that newly labeled data to train models for the named object categories. Given a novel image region, it predicts the object label.

We train one-vs-one SVM classifiers (with $C = 1$) for all discovered categories using the appearance kernels. To simulate obtaining labels from a human annotator, we label all instances in a cluster according to the ground-truth majority instance. In addition to the baselines from above, we compare to two "upper bounds" in which the ground truth labels on all instances are used to train a nearest-neighbor (NN) and SVM classifier. We test on the 40% split that trained the stuff models (which is fine, since the test set used here consists only of objects), totaling 2,836 test windows from 16 object categories.

Table 1 shows the results, for a range of iterations. Alongside test accuracy, we show the number of manually-provided labels required by each method. As expected, the fully supervised methods provide the highest accuracy, yet at the cost of significant human effort (one label per training window). On the other hand, our method requires a small fraction of the labels (one per discovered category), yet still achieves accuracy fairly competitive with the supervised methods, and substantially better than either the batch or hardest-first baselines.

This result suggests a very practical application for discovery, since it shows that we can greatly reduce human annotation costs and still obtain reliable category models.

Conclusions: We introduced a self-paced discovery framework that progressively accumulates object models from unlabeled data. Our experiments demonstrate its clear advantages over traditional batch approaches and representative state-of-the-art techniques. In future work, we plan to explore related ideas in the video domain, and further investigate how such a system can most effectively be used for interactive labeling with a human-in-the-loop.

	Ours	Hardest first	Batch	Sup. NN	Sup. SVM
# of labels required	10	10	10	2721	2721
accuracy (%)	47.71	27.33	33.96	54.69	64.39
# of labels required	20	20	20	—	—
accuracy (%)	47.14	26.16	34.34	—	—
# of labels required	30	30	30	—	—
accuracy (%)	45.80	29.16	29.90	—	—
# of labels required	40	40	40	—	—
accuracy (%)	49.15	27.19	32.51	—	—

Table 1. Classification results on novel images, where discovered categories are interactively labeled. Our approach yields good prediction accuracy for minimal human effort.

Acknowledgements: We thank the CALVIN group for sharing their objectness code, and Alyosha Efros, Joydeep Ghosh, and Jaechul Kim for helpful discussions. This research is supported in part by DARPA CSSG N10AP20018 and NSF EIA-0303609.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an Object? In *CVPR*, 2010.
[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From Contours to Regions: An Empirical Evaluation. In *CVPR*, 2009.
[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum Learning. In *ICML*, 2009.
[4] I. Endres and D. Hoiem. Category Independent Object Proposals. In *ECCV*, 2010.
[5] G. Heitz and D. Koller. Learning Spatial Context: Using Stuff to Find Things. In *ECCV*, 2008.
[6] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised Modeling of Object Categories Using Link Analysis Techniques. In *CVPR*, 2008.
[7] B. Kuipers, P. Beeson, J. Modayil, and J. Provost. Bootstrap Learning of Foundational Representations. *Connection Science*, 18(2), 2006.
[8] M. P. Kumar, B. Packer, and D. Koller. Self-Paced Learning for Latent Variable Models. In *NIPS*, 2010.
[9] S. Lazebnik and M. Raginsky. An Empirical Bayes Approach to Contextual Region Classification. In *CVPR*, 2009.
[10] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning from Partially Matching Images. *IJCV*, 85(2), May 2009.
[11] Y. J. Lee and K. Grauman. Object-Graphs for Context-Aware Category Discovery. In *CVPR*, 2010.
[12] L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: Automatic Object Picture CollecTion via Incremental mOdel Learning. In *CVPR*, 2007.
[13] D. Liu and T. Chen. Unsupervised Image Categorization and Object Localization using Topic Models and Correspondences between Images. In *ICCV*, 2007.
[14] T. Liu, J. Sun, N.-N. Zheng, X. Tang, and H.-Y. Shum. Learning to Detect A Salient Object. In *CVPR*, 2007.
[15] T. Malisiewicz and A. Efros. Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In *NIPS*, 2009.
[16] E. Olson, M. Walter, J. Leonard, and S. Teller. Single Cluster Graph Partitioning for Robotics Applications. In *RSS*, 2005.
[17] P. Perona and W. Freeman. A Factorization Approach to Grouping. In *ECCV*, 1998.
[18] M. Ranzato, Y. Boureau, and Y. LeCun. Sparse Feature Learning for Deep Belief Networks. In *NIPS*, 2007.
[19] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.
[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *ECCV*, 2006.
[21] Tan, Steinbach, and Kumar. *Introduction to Data Mining*. 2005.
[22] Z. Tu. Auto-context and Application to High-level Vision Tasks. In *CVPR*, 2008.
[23] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised Object Discovery: A Comparison. *IJCV*, 2010.