Efficient Label Collection for Unlabeled Image Datasets

Maggie Wigness, Bruce A. Draper and J. Ross Beveridge
Colorado State University
Fort Collins, CO
mwigness,draper,ross@cs.colostate.edu

Abstract

Visual classifiers are part of many applications including surveillance, autonomous navigation and scene understanding. The raw data used to train these classifiers is abundant and easy to collect but lacks labels. Labels are necessary for training supervised classifiers, but the labeling process requires significant human effort. Techniques like active learning and group-based labeling have emerged to help reduce the labeling workload. However, the possibility of collecting label noise affects either the efficiency of these systems or the performance of the trained classifiers. Further, many introduce latency by iteratively re-training classifiers or re-clustering data. We introduce a technique that searches for structural change in hierarchically clustered data to identify a set of clusters that span a spectrum of visual concept granularities. This allows us to efficiently label clusters with less label noise and produce high performing classifiers. The data is hierarchically clustered only once, eliminating latency during the labeling process. Using benchmark data we show that collecting labels with our approach is more efficient than existing labeling techniques, and achieves higher classification accuracy. Finally, we demonstrate the speed and efficiency of our system using real-world data collected for an autonomous navigation task.

1. Introduction

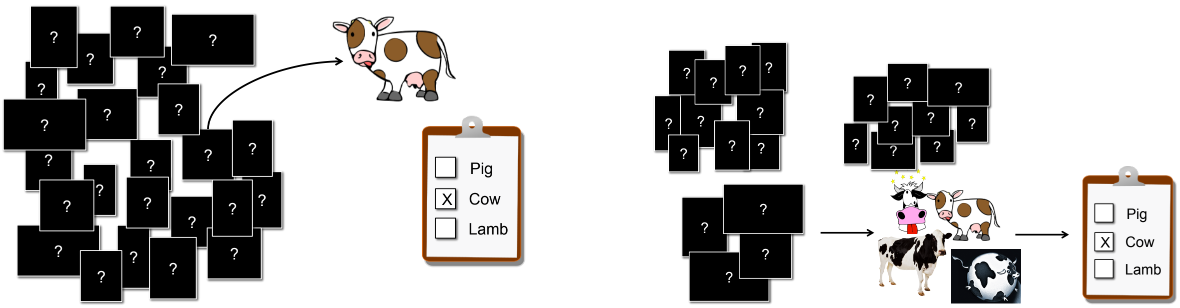
Classification is an important task for many visually intelligent systems, but there are a variety of challenging properties associated with visual data that make it difficult. Some of these challenges include changes in illumination, scale, perspective, color and background clutter. Classifiers try to account for these factors, but often need large amounts of training data to learn these variations. While collecting visual data is a trivial task, the raw data itself contains no label information for training supervised classifiers.

Label collection is a burdensome task for human annotators, and unfortunately may not be a one time event. For example, military robots may be constantly moving to new domains where new training data must be collected (see Section 5). Fortunately, efficient labeling schemes have emerged to help alleviate some of the labeling workload, while still producing sets of labeled data capable of training high performing classifiers. In the context of this paper, efficiency is defined relative to the workload required from a fully supervised system, *i.e*., hand labeling each image in the training set individually.

Active learning [9, 10, 11, 12, 15] reduces the workload by requiring only a subset of the most informative images in the training data to be labeled. The label collection process and classifier performance are tightly coupled in these frameworks since classifiers are re-trained on each iteration to help select the next samples to label. Like a fully supervised approach, active learning applies labels at the level of individual data samples (seen in Figure 1(a)). Thus, these frameworks are only efficient if a subset of the unlabeled images can sufficiently train classifiers.

Group-based labeling [4, 14, 16, 18, 19] is potentially more efficient than instance-based labeling because a single label is assigned to a group of images simultaneously (as in Figure 1(b)). Unlike active learning, group labeling can be a noisy strategy if a group contains data from multiple visual concepts. This may occur when data are grouped using feature patterns that represent a concept broader than the classifier label set. Assigning the dominating class label trades some label accuracy for efficiency, but some techniques avoid label noise all together. Examples include collecting binary constraints to iteratively improve clustered output [2, 22], removing images from a group that do not match the dominating label [7] or only collecting labels from groups that represent exactly one concept [21]. All of these approaches result in more effort or latency to assign noise-free labels.

This paper introduces a group-based labeling technique that balances the trade-off between efficiency and label accuracy more effectively than previous techniques. Our approach uses hierarchical clustering to establish a space of groupings across a spectrum of visual concept granulari-



(a) Instance-based labeling

(b) Group-based labeling

Figure 1. Illustration of two different labeling approaches given a set of unlabeled images.

ties. By maintaining the hierarchy, our system can search for groups that match the concept granularity of the classifier and thereby keep label noise to a minimum. These groups are identified by searching for large local structural changes in the hierarchy. Overall, our labeling framework identifies a subset of clusters from the hierarchy that can be labeled with little effort, produce minimal label noise and train high performing classifiers.

Using two benchmark datasets we show that our hierarchical cluster guided labeling approach is more efficient than state of the art labeling approaches and achieves higher classification accuracy. Further, we demonstrate the speed and feasibility of our labeling system in a real-world scenario. Since no latency is introduced during the labeling process, a single human annotator can label new training data in less than 45 minutes for an autonomous navigation task.

2. Background

Many techniques have been introduced to address the labeling workload problem. In the context of this paper, we discuss the labeling process as applying an object or scene class label to unlabeled data samples. A data sample may be an image from a dataset or a region from within an image, but each sample is assigned a single ground truth label. Note that the same techniques could be used to assign labels to videos or other types of data.

Labeling interactions look different for each technique, but a definition of labeling effort applicable to all techniques needs to be established to make direct comparisons. The remainder of this section discusses existing labeling techniques and how labeling effort is computed for each.

2.1. Labeling effort

In this paper, the task is to collect labels for supervised classifiers. Without an efficient labeling technique, the total effort of a fully supervised approach would be to provide a label to every training sample. Thus, the effort required to label a dataset with n training samples is defined as:

$$\text{Labeling Effort} = \frac{\#\text{ interactions}}{n} \tag{1}$$

An interaction is different for every labeling system, but is an overall representation of the number of times a human annotator provides information to the system. The following types of interactions are used in techniques that address labeling workload:

1. Providing a class label for an image [9, 10, 11, 12, 15]
2. Providing a class label for a group of images [4, 7, 13, 16, 18, 19, 21]
3. Indicating that a group lacks a common label [21]
4. Removing an image from a group [7]
5. Indicating whether or not two images represent the same label [2, 22]

We note that not all of the interactions in the list have the same cognitive load. However, an in-depth analysis of cognitive differences and their impact on labeling efficiency is beyond the scope of this paper. While we treat each interaction equally when discussing labeling effort, we will discuss some of the major performance differences with respect to cognitive load in the comparisons made in Section 4.

2.2. Related work

There are two dominating approaches used to address the labeling workload problem: active learning and group-based labeling. Active learning tries to identify the most informative subset of training samples to train classifiers. Selection criteria includes probabilistic uncertainty sampling [9, 10, 11], Gaussian process models [12] and information density [15]. Active learning reduces the number of labels provided by a user, but may require a priori knowledge for classifier seeding and introduces latency while iteratively re-training classifiers.

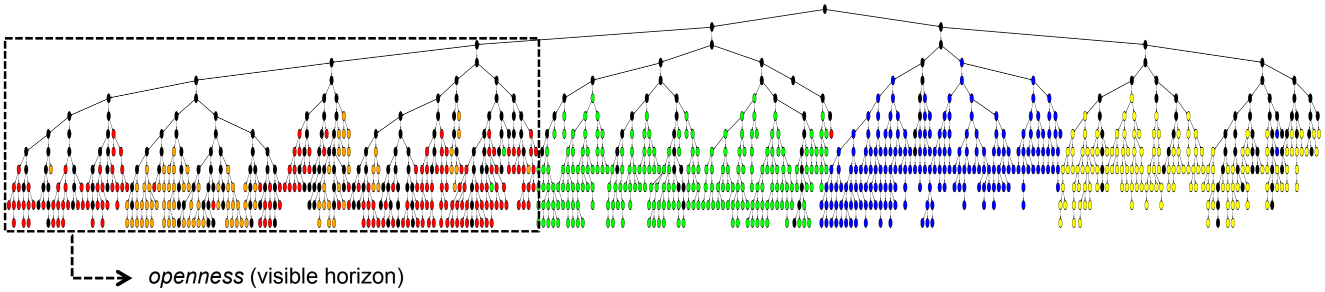


Figure 2. Hierarchical clustering of five classes from the 13-Scenes dataset. Dotted line highlights the data grouped as the coarse-grained visual concept *openness*.

Group-based labeling provides a single label to a group of samples. Clustering [4, 14, 19] and topic modeling [16, 18] form groups through bottom-up discovery, requiring no a priori knowledge of the unlabeled data. These techniques try to find a one-to-one mapping between groups and visual concepts. Unfortunately, visual data properties make grouping difficult and groups often contain data from multiple classes. Assigning the dominating class label to the entire set of images can create significant label noise.

Label noise has been reduced at the cost of additional labeling effort and labeling latency. Active clustering improves group coherency by iteratively collecting constraints to augment feature representation [2, 8, 22]. Lee and Grauman cluster the "easiest" subset of unlabeled data and label a single group at each iteration to improve overall group coherency [13]. Largest subset labeling removes all label noise by asking a user to remove images from a group that do not represent the dominating class label [7].

One oversight of existing group-based techniques may be their lack of exploitation of the hierarchical semantics that visual data exhibits. Hierarchical labels appear in many related visual problems. Taxonomies of image datasets, collected via crowdsourcing, display hierarchical semantics [3]. Also, Deng et al. use label semantics to define binary hierarchical queries to efficiently label the existence of visual concepts in multi-label images [5]. In previous work, we used hierarchical clustering to relax the one-to-one grouping constraint, but the hierarchical structure only served as a basis to form a set of groups and was never fully exploited [21]. Specifically, this technique focused on noise-free labeling by incrementally training a coherency model on-line, using previously labeled groups and their associated stability measure. The model predicted the likelihood that groups represented a single visual concept, but only the selected groups that were 100% coherent received labels.

This paper differs from existing techniques in many ways. First, it purposefully seeks to identify groups that span multiple concept granularities, which are encoded in the hierarchical clustering. Second, it exploits the structural relationships in the hierarchy to select groups to label. Finally, it avoids labeling latency caused by re-training, re-clustering and online-modeling.

3. Hierarchical cluster guided labeling

Our labeling system is designed to be quick and efficient so new sets of labeled training data can be collected by a single human annotator in mere hours or less. Our approach, called hierarchical cluster guided labeling (HCGL), labels groups of images to train classifiers. HCGL selects a subset of groups from a hierarchical clustering of unlabeled data covering a range of visual concept granularities.

One disadvantage of group-based labeling is the addition of label noise when images in the same group represent multiple visual concepts. As discussed before, one reason this can occur is that similarities come in a range of granularities. For example, grouping images of *coast* and *highway* together makes sense at a very coarse-granularity because these scenes share an *openness* quality since the horizon is visible. Also, images of *dog*, *cow* and *sheep* share a coarser-grained label of *animal*. Groupings are influenced by feature representation, intra-class and inter-class similarity, which are hard to manage explicitly with a flat partitional grouping algorithm.

Instead of forcing groupings to occur at a particular level of granularity, we use hierarchical clustering to maintain a spectrum of image groupings. Figure 2 illustrates this concept with a hierarchical clustering of five classes from the well known 13-Scenes dataset [6]. Nodes colored black correspond to groups that contain images from multiple scene classes. The remaining colors indicate groups of images from a single scene class. There is a natural division of the hierarchy into four groups: *tall building* (green), *living room* (blue), *suburb* (yellow) and the coarse-grained concept of *openness* (dashed outline) previously mentioned. The many smaller, inter-weaved partitions of the *coast* (red) and *highway* (orange) classes is evidence of high inter-class similarity. By maintaining the hierarchical structure, we can search for locations in the hierarchy that coherently correspond to the classifier label set, and thereby reduce label noise.

The data hierarchy, denoted as $\mathcal{H}$, is redundant in the sense that if a group's samples are from a single class, its descendants (which are subgroups) inherit that class label. Therefore, not every group in the tree needs to be labeled. Instead, we select a subset of groups from $\mathcal{H}$ that represent possible candidate concepts. In Figure 2, this includes groups that represent scene categories, and groups that represent concepts such as *openness* or *outside*. HCGL selects these groups by looking at the local structural changes induced by splits in the hierarchy. This selection forms a set of unordered groups, $\mathcal{S}$, where $\mathcal{S} \subset \mathcal{H}$. After selecting $\mathcal{S}$, an ordering of the groups is established for labeling.

3.1. Modeling structural change

To keep $\mathcal{S}$ concise, only groups formed from significant structural changes, after a split in $\mathcal{H}$, are added to the subset. Structural change is used as an indicator of a change in visual concept. Specifically, we suggest that the dominate direction of variance of a group of data, in a high dimensional feature space, provides information about the underlying structure and corresponding concept of the samples. When there is a change in the direction of variance, the underlying concept of a group may also change.

We represent the internal structure of a group in $\mathcal{H}$ by the eigenvectors of the covariance matrix of its samples. Local structural change is found by comparing the internal structure of a group, c, to one of its ancestors. In this paper, the comparison is modeled as the angle between c, and its parent, p (relationship seen in Figure 3). Specifically, the angle between the first eigenvector of c and p, denoted as v_c and v_p respectively. Formally, structural change for group c is defined as the cosine distance,

$$\Delta(c) = 1.0 - |\cos(\langle v_c, v_p \rangle)|, \tag{2}$$

which yields values in the interval of $[0.0, 1.0]$, where large Δ values represent large angles.

Most groups in $\mathcal{H}$ have at least some structural difference from their parent, but $\mathcal{S}$ should represent only the splits that are likely to result from a change of concept. To detect these transitions, HCGL looks for large changes in structure followed by a lack of structural change in local neighborhoods of $\mathcal{H}$. In other words, if the structural change of c is a local peak with respect to p and its children, c_r and c_l (relationship illustrated in Figure 3), it is added to $\mathcal{S}$. Formally, $\mathcal{S}$ contains any c that satisfies the following two conditions:

$$\begin{aligned} \Delta(c) &> \Delta(p) \\ \Delta(c) &> \Delta(c_{r,l}) \end{aligned} \tag{3}$$

$\mathcal{S}$ has two important properties. First, groups in $\mathcal{S}$ are not necessarily disjoint because every image belongs to many related groups in the hierarchical structure. Second, selecting peaks in structural change does not guarantee that every image will be represented in $\mathcal{S}$. We discuss the first property in the context of the group labeling order in the next section. The second property may result in only a fraction of the training data receiving labels, which is analyzed during experimental evaluation.

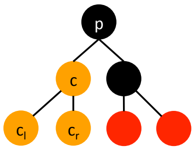


Figure 3. Illustration that depicts the relationships for group c in a local neighborhood of $\mathcal{H}$, including its parent p and left and right children, c_l and c_r.

3.2. Group labeling

Flat partitional grouping forms a set of disjoint groups, labels each group, and then trains a classifier with the collected data. HCGL is different because groups in $\mathcal{S}$ are not necessarily disjoint. The ordering of groups in $\mathcal{S}$ is meaningful because if a group is given a class label, all descendants of this group (according to the structure in $\mathcal{H}$) inherit that label, and thus no longer need to be labeled by the annotator.

There are many ways $\mathcal{S}$ can be ordered. Since groups have already been selected as meaningful based on their structural change, by default HCGL ranks groups in descending order by their Δ value. The idea is to order groups by the strength of their potential concept transition. During labeling, when a group is given a class label, any of its descendants that exist in $\mathcal{S}$ are removed since they inherit the label. Thus, the total labeling effort of HCGL is not equal to $|\mathcal{S}|$, but depends on the labeling order and the number of inherited labels. The entire HCGL process is outlined in Algorithm 1.

Algorithm 1 Hierarchical Guided Cluster Labeling

```
Require: $\mathcal{H}$
$S = \{\}$
for all $c \in \mathcal{H}$ do
    $relatives = \{p, c_l, c_r\}$
    if $\Delta(c) > \Delta(r)$ , $\forall\ r \in relatives$ then
        $\mathcal{S} = \mathcal{S} \cup \{c\}$
$\mathcal{S} = sort(\mathcal{S}, \Delta)$
while $\mathcal{S} \neq \emptyset$ do
    label querying $\rightarrow \mathcal{S}[0]$
    $update(S)$
```

As mentioned earlier, groups selected for $\mathcal{S}$ represent a range of visual concept granularities. At the time of labeling, a group of images may be shown to an annotator that represents a concept broader than the classifier label set. The human annotator will recognize that the concept represented by the group is not relevant to the classifier and continues to the next selected group of images. To simulate this in our automated labeling experiments, if more than 50% of a group's images represent a single visual concept, it is given the dominating class label. When a group does not have a majority of images that map to a single label, the group is given an *irrelevant* label. This query counts towards the total level of effort, but does not provide additional labels for the classifier.

4. Benchmark data evaluation

To evaluate HCGL, we make several direct comparisons to state of the art labeling techniques on two benchmark datasets. These comparisons evaluate how quickly each technique collects labeled data and how effectively the data trains a supervised classifier. Classification accuracy on a disjoint test set is computed iteratively after each labeling query. It is easy to simulate human interactions automatically with benchmark datasets using the available ground truth, and automating this process allows systems to run to completion. However, we focus our evaluation on the performance achieved in the earlier stages of labeling effort to acknowledge that labeling resources may be scarce in real-world scenarios. For large unlabeled datasets, it may not be feasible to provide the amount of labeling effort that automated experiments provide.

Comparisons to HCGL are made using a set of diverse labeling frameworks that require different types of labeling interactions. These methods include:

- **SAC** (Spectral Active Clustering [22]) - active clustering approach that queries for 20 binary constraints per iteration to improve one-to-one clustered output, followed by majority labeling
- **SG** (Selective Guidance [21]) - hierarchical clustering approach that models group coherency for selection and only labels groups that contain images from exactly one class
- **MKML** (Multiple Kernel Metric Learning [7]) - iterative clustering approach that labels the largest subset of samples with the dominating class and removes images that do not match this label

These techniques are compared and analyzed using classification accuracy versus labeling effort results on one scene and one object dataset. For all experiments, HCGL builds $\mathcal{H}$ using agglomerative clustering with Ward's linkage [20] and Euclidean distance. Each experiment is averaged over 10 trials of random training/testing partitions. Finally, classification performance of a fully supervised labeling approach is also computed to indicate the upper bound on performance.

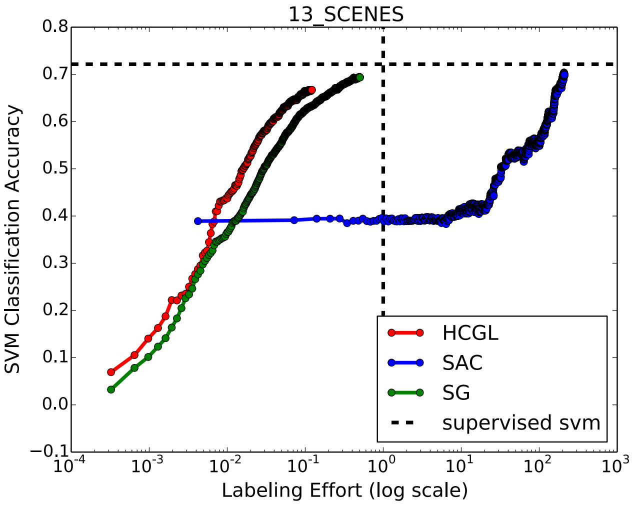


Figure 4. Comparison of classification accuracy versus labeling effort on the 13-Scenes dataset.

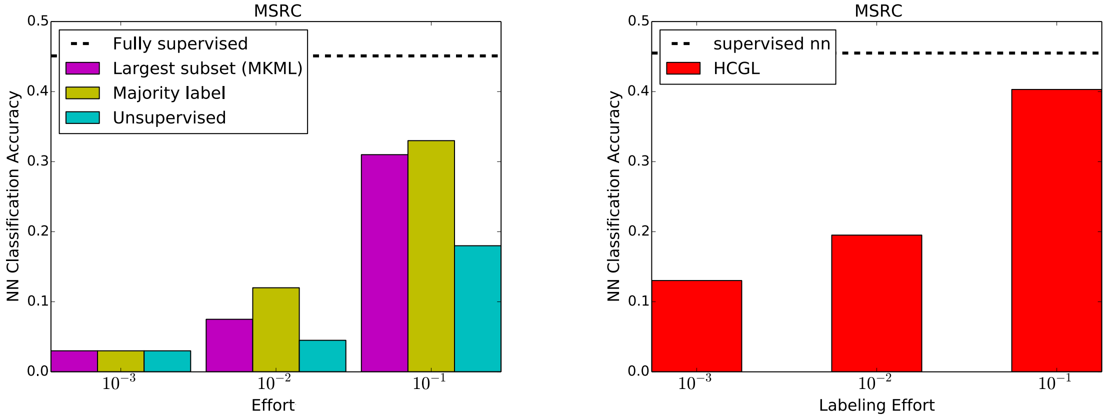
4.1. Scene labeling and classification

For the first experiment, we use the 13-Scenes dataset [6] that is comprised of images representing 13 natural scene classes. GIST descriptors [17] are used to represent each image, and an 80/20 partition is used to divide the data into training and testing sets, respectively. For classification, we train an SVM classifier using the same parameters found in existing efficient labeling literature [11, 21].

Using code made available by the authors, we directly compare the performance of HCGL to that of SAC and SG. Results of this experiment can be seen in Figure 4. With only one-tenth of the effort needed to fully label the dataset, HCGL outperforms the other techniques and approaches supervised performance. SAC requires a significant amount of effort before any improvements are made and reasonable classification performance is achieved. In fact, the labeling effort exceeds the effort required to fully label the dataset (indicated by the vertical dashed line). Recall however, that SAC queries for binary constraints, which are cognitively easy to answer but only provide a single bit of information. Thus, it is not surprising that many binary queries are required. Eventually SG reaches a higher classification performance than HCGL, but this marginal performance boost comes at the cost of about three times more effort.

4.2. Object labeling and classification

Our second experiment replicates the experimental protocol used in the MKML paper [7] on the MSRC dataset.



(a) Extracted MKML results

(b) Corresponding HCGL results

Figure 5. Comparison of classification accuracy for three levels of labeling effort on the MSRC dataset.

This dataset includes images of objects from 21 different classes. In the original experiment, the authors use a 40/60 data partition. The 40% split was used to extract regions and features representing 5 classes that were presumed known to act as a seed to their system. HCGL assumes no known knowledge while collecting labels and therefore does not use this data for seeding. The other 60% is used to perform the grouping and label collection for the remaining 16 unknown classes. Using the collected labels, classification is performed only on the 16 classes that were presumed unknown, which means the 40% split can also be used at the testing set.

Each image in the MSRC dataset contains multiple objects to be labeled, so the images are first segmented into regions. Each region is treated as a separate data sample that represents a single visual concept. We use the publicly available segmentation and appearance feature extraction code used by Lee and Grauman [14] to generate regions for the MSRC dataset. While the image region set is not identical to the set used by MKML, we achieve the same supervised nearest neighbor classification performance as MKML, indicating that the sets of training data are effectively equivalent.

A comparison between HCGL and MKML can be seen in Figure 5. The MKML results (Figure 5(a)) are an alternative view of the authors' original presentation (Figure 10 [7]). The three bars correspond to their proposed largest subset labeling technique, a majority labeling technique intended to emulate an incremental labeling system [13] and an unsupervised baseline. Further, as mentioned earlier the focus of comparison is on the classification results achieved during the earliest stages of labeling effort.

The results are separated in side by side plots because the definition of labeling effort used by MKML is slightly different than what is defined in Section 2.1. In particular, MKML defines effort as the fraction of images that are removed from a group because they do not match the largest subset label (interaction 4 in Section 2.1). It does not include the effort required to provide the label of the largest subset.

When focusing on results achieved with minimal labeling effort, the majority labeling technique in Figure 5(a) outperforms MKML with largest subset labeling, which is noted by the authors. However, the majority labeling technique still performs significantly worse than the fully supervised approach. HCGL uses a similar majority labeling scheme, but outperforms all techniques from the MKML paper. The performance gap suggests that groups in $\mathcal{S}$ are more coherent than those selected in the MKML approach since label noise impacts classification performance or requires more labeling effort for the MKML largest subset labeling approach.

4.3. Secondary evaluation criteria

Section 4.1 and 4.2 show that HCGL collects labels and trains higher performing classifiers with less effort than existing labeling techniques. We are interested in whether the performance of HCGL is primarily a result of the selection of groups to form $\mathcal{S}$, or a result of the ordering in which groups of $\mathcal{S}$ are labeled. To probe further, we investigate other evaluation criteria, and compare different orderings of $\mathcal{S}$ on the 13-Scenes dataset. Unlabeled groups are iteratively selected using three selection criteria:

- **HCGL-Δ** - maximum structural change
- **HCGL-Size** - maximum number of unlabeled samples
- **HCGL-Path Length** - maximum path length to a labeled group

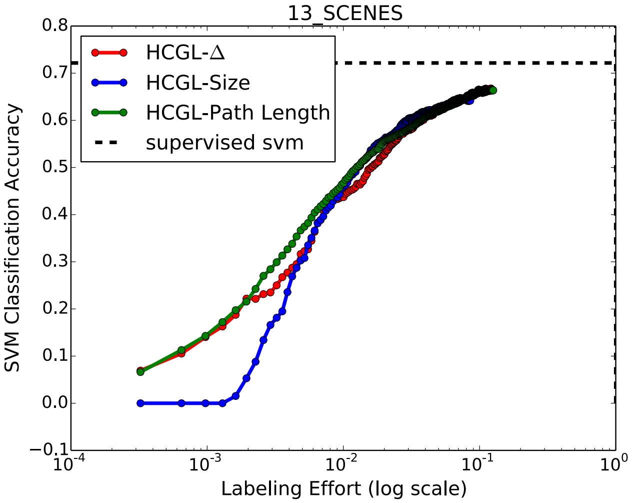
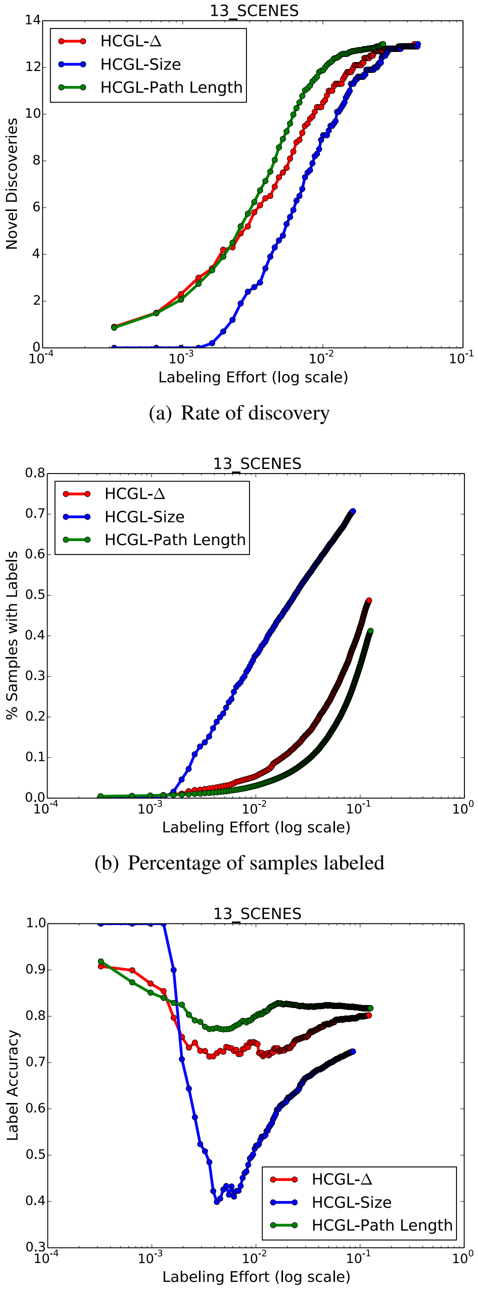


Figure 6. Comparison of classification accuracy versus labeling effort on the 13-Scenes dataset for three orderings of $\mathcal{S}$.

HCGL-Δ is the original ordering described in Section 3.2 that ranks groups by the likelihood of visual concept transition. Size and path length orderings are introduced to emphasize other labeling goals. Ordering by size emphasizes the efficiency at which labels are assigned. Ordering by path length emphasizes novel concept discovery by selecting groups spread throughout the hierarchy. Unlike the other two criteria, path length ordering is dependent on previous selections. Briefly, every unlabeled group is represented by the shortest path length between it and the previously labeled groups. These path lengths are then used to select the group that is least similar to what has already been discovered and labeled.

Figure 6 shows the classification performance for the three ordering criteria. The only major performance difference is seen very early in the labeling process when HCGL-Size performs significantly worse the other two criteria. This slow performance start is a result of selecting large groups representing a concept too broad to be labeled, which receive an *irrelevant* label. On the whole, we conclude that the success of HCGL is primarily a result of the selections made to build $\mathcal{S}$ and less about the particular labeling order.

However, as a secondary form of analysis we present other evaluation criteria that shows the emphasis of the three ordering techniques are in fact different. These evaluations relate to qualities that training data should probably possess. First, labeled samples should span the concept label set. Second, there should be a sufficient number of samples labeled. Finally, samples should be labeled accurately. Figure 7 compares the different orderings for these evaluation criteria.



(a) Rate of discovery

(b) Percentage of samples labeled

(c) Label accuracy

Figure 7. Secondary evaluation criteria of different labeling techniques.

Discovery of concepts is important for group-based labeling methods because most do not start with seed sets, and classifiers can only recognize concepts that exist in the labeled training data. Techniques that collect labels for all k classes the fastest will likely see the fastest classification performance boost. As designed, Figure 7(a) shows that ordering by path length provides the best rate of discovery, followed by Δ ordering. This emphasis on discovery is also important in explaining the performance gap seen in the classification results.

Figure 7(b) reinforces that labeling all of $\mathcal{S}$ does not guarantee that HCGL assigns a label to all training samples. Again as expected, ordering by size produces more labeled samples faster than other orderings after it gets past its initial selections that are too broad to label. However, the trade-off between efficiency and label accuracy is apparent when also looking at Figure 7(c). Label accuracy is the fraction of noiseless labels collected. Ordering by size maintains perfect accuracy for the queries where no label is provided, but results in the lowest label accuracy of all orderings once it begins assigning labels.

Interestingly, these evaluations suggest that training data does not need to be perfectly accurate or labeled in its entirety to achieve high classification performance. While we do not claim that label noise has zero impact on classification accuracy, it does not seem to degrade performance significantly. This may be because the label noise introduced by HCGL is not random. Label noise enters the system because data from different concepts share a similarity, which is why they are grouped together. Overall, emphasizing different goals during the labeling process results in a set of labels that captures the essence of $\mathcal{S}$ so all techniques achieve similar classification results.

5. Labeling speed evaluation

Up to this point, the focus of evaluation has been on performance with respect to the number of human interactions during the labeling process. Since the previous experiments could be automated with benchmark data, the evaluation says very little about the amount of time required for a human to interact with the system. This interaction time would include the latency introduced by many techniques that recluster data or re-train classifiers.

To demonstrate the speed and real-world feasibility of HCGL, we present the results of labeling real-world data collected to train classifiers for autonomous robot navigation. This experiment is motivated by military applications that frequently send robots into new domains with different terrains or environments. Training a classifier to learn the terrains and objects in the environment needs to happen quickly with relatively low operator interaction.

In collaboration with a department of defense agency, images were collected of the premises of a military training facility from a small autonomous robot. Each image may contain several terrains (*e.g.*, *grass* and *dirt*) or objects (*e.g.*, *buildings* and *trees*). An example image can be seen in Figure 8(a). Since images contain multiple concepts of interest, each image is over-segmented using SLIC [1], resulting in a total of 5,951 segments to be used as training data.

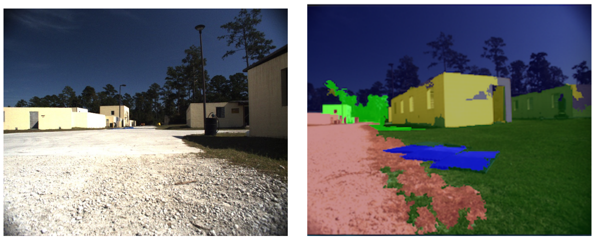

(a) Training image (b) Classification results

Figure 8. Example images from a real-world labeling scenario.

For this simple demonstration, a single human annotator spent less than 45 minutes labeling this training set. We are unable to provide a quantitative evaluation of classification performance because there is no ground truth associated with this data. However, a qualitative evaluation of classification using test images suggests reasonably high performance. Figure 8(b) is an example of the classification results for a test image. Each color indicates a different visual concept. Although the classification is not perfect, the results are believed to be strong enough for the navigation task at hand. In very short order enough labels were collected for the classifier to recognize eight unique terrain and object classes.

6. Conclusion

Visual classifiers are often domain dependent. Collecting data for a new domain is trivial, but attaching labels to this data requires significant human effort. Many techniques have emerged to help reduce the labeling workload. However, these systems often require a priori knowledge for seeding, introduce latency or struggle with how to handle the collection of label noise. We have presented a hierarchical cluster guided labeling (HCGL) system that maintains groups of images that represent all possible candidate concepts in the data. By maintaining all possible grouping granularities, HCGL can label groups that coherently match the classifier label set so minimal label noise is introduced. Groups of images that represent a concept that is too coarse for use by a classifier are easily identified by an annotator and are marked as irrelevant with very little added effort. Our group selection method allows HCGL to collect labels with less effort than existing approaches and produce higher performing classifiers as well. We also demonstrated that HCGL is feasible to use in real-world scenarios that require fast collection of training labels with few labeling resources.

Acknowledgments

This research has been supported by a cooperative agreement (W911NF-10-2-0066) with US Army Research Laboratory.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 8

[2] A. Biswas and D. Jacobs. Active image clustering: Seeking constraints from humans to complement algorithms. In *Proceedings of Computer Vision and Pattern Recognition*, pages 2152–2159. IEEE, 2012. 1, 2, 3

[3] L. B. Chilton, G. Little, D. Edge, D. S. Weld, and J. A. Landay. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2013. 3

[4] D. Dai, M. Prasad, C. Leistner, and L. Van Gool. Ensemble partitioning for unsupervised image categorization. In *Proceedings of European Conference on Computer Vision*, pages 483–496. Springer, 2012. 1, 2, 3

[5] J. Deng, O. Russakovsky, J. Krause, M. S. Bernstein, A. Berg, and L. Fei-Fei. Scalable multi-label annotation. In *Proceedings of Human Factors in Computing Systems*, pages 3099–3102. ACM, 2014. 3

[6] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 524–531. IEEE, 2005. 3, 5

[7] C. Galleguillos, B. McFee, and G. Lanckriet. Iterative category discovery via multiple kernel metric learning. *International Journal of Computer Vision*, 108(1-2):115–132, 2014. 1, 2, 3, 5, 6

[8] A. Gilbert and R. Bowden. igroup: Weakly supervised image and video grouping. In *Proceedings of International Conference on Computer Vision*, pages 2166–2173, 2011. 3

[9] A. Holub, P. Perona, and M. C. Burl. Entropy-based active learning for object recognition. In *Proceedings of Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008. 1, 2

[10] P. Jain and A. Kapoor. Active learning for large multi-class problems. In *Proceedings of Computer Vision and Pattern Recognition*, pages 762–769. IEEE, 2009. 1, 2

[11] A. J. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Proceedings of Computer Vision and Pattern Recognition*, pages 2372–2379, 2009. 1, 2, 5

[12] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active learning with gaussian processes for object categorization. In *Proceedings of International Conference on Computer Vision*, pages 1–8. IEEE, 2007. 1, 2

[13] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1721–1728. IEEE, 2011. 2, 3, 6

[14] Y. J. Lee and K. Grauman. Object-graphs for context-aware visual category discovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):346–358, 2012. 1, 3, 6

[15] X. Li and Y. Guo. Adaptive active learning for image classification. In *Proceedings of Computer Vision and Pattern Recognition*, 2013. 1, 2

[16] D. Liu and T. Chen. Unsupervised image categorization and object localization using topic models and correspondences between images. In *Proceedings of International Conference on Computer Vision*, pages 1–7. IEEE, 2007. 1, 2, 3

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. 5

[18] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proceedings of International Conference on Computer Vision*, volume 1, pages 370–377 Vol. 1, Oct 2005. 1, 2, 3

[19] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88(2):284–302, 2010. 1, 2, 3

[20] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. 5

[21] M. Wigness, B. A. Draper, and J. R. Beveridge. Selectively guiding visual concept discovery. In *Proceedings of the Winter Conference on Applications of Computer Vision*. IEEE, 2014. 1, 2, 3, 5

[22] C. Xiong, D. M. Johnson, and J. J. Corso. Spectral active clustering via purification of the k-nearest neighbor graph. In *Proceedings of European Conference on Data Mining*, 2012. 1, 2, 3, 5