Abstract

Labeling data to train visual concept classifiers requires significant human effort. Labeling overhead has been addressed with active learning, but often these approaches assume a fixed set of visual concepts, i.e., a closed-world. Unsupervised concept discovery is designed to learn concepts bottom-up by grouping, thereby reducing the labeling workload by moving from instance-based to group-based labeling. However, many of these approaches still operate in a closed-world, and fail to discover coherent groups that represent all concepts. We introduce a hierarchical clustering based approach to discover visual concepts that generalizes to unlabeled open-world data sets. We selectively guide the labeling process by searching the hierarchy for clusters that are believed to coherently represent a single object class. Using benchmark data sets we discuss the discovery performance using our selective guidance approach, and show that selective guidance collects labeled training data that achieves classification accuracy better than active learning approaches with less annotation workload.

1. Introduction

Supervised multi-class visual concept classifiers require a significant amount of labeled training data to yield high classification accuracy. Although the collection of visual data has become trivial, the task of labeling these large sets of data requires significant human effort. Thus, unsupervised and semi-supervised approaches are emerging with the goal of training an accurate classifier while significantly reducing the labeling workload.

It has been shown that with active learning, fewer training samples can be used without compromising the performance of visual concept classification [6, 11, 8, 5, 7]. Active learning techniques iteratively select samples from a pool of unlabeled data to find a subset of diverse samples that achieve high classification accuracy. Selection of the subset is typically based on the classification uncertainty of the unlabeled data.

Even though active learning reduces the number of samples needed for training, most of these frameworks assume that the number of visual concepts in the training set is known in advance. Further, many of these frameworks also assume that a set of labeled samples from each of the visual concepts is readily available. These assumptions limit most active learning frameworks to closed-world problems.

Visual concept discovery is another approach that has emerged in part to help minimize the need to label data. Visual concept discovery algorithms often rely on clustering [17, 1, 9] or latent topic models [12, 16] to form groups of images that represent concepts. These approaches reduce the labeling workload by learning one-to-one mappings between groups and visual concepts. This requires only a single label per visual concept from an annotator.

Unfortunately, achieving a perfect partition of visual data has proven to be difficult. The variations in intra-class and inter-class similarity make some concepts easy to group while others are incredibly challenging. Further, even though visual concept discovery is designed to learn concepts bottom-up, many existing algorithms still require parameters that define the number of visual concepts to discover. For example, specifying the \( k \) in k-means clustering. Once again, this limits these techniques to closed-world problems.

This paper presents an algorithm that discovers visual concepts without any a priori knowledge of the data, and efficiently labels these concepts to create training data that achieves high classification accuracy. We do this by hierarchically clustering data to create a cluster-space in which to search for visual concepts. Clusters are iteratively selected for labeling by evaluating the likelihood of information gain in terms of exploitation, i.e., collecting labels for a large number of samples, and exploration, i.e., the discovery of new visual concepts. We use benchmark data sets to show that with selective guidance visual concepts can be discovered bottom-up, and that the collected set of labeled samples achieves classification accuracy better than existing active learning frameworks with fewer labeling queries.
2. Background

This section discusses existing techniques that help minimize the demanding labeling workload of visual data. We begin with a discussion of active learning frameworks, followed by unsupervised concept discovery techniques.

2.1. Active Learning Frameworks

Reducing the high cost of annotation while maintaining high classification accuracy is the primary goal for active learning frameworks. Pool-based frameworks divide data into three sets: 1) seed set - an initial set of labeled instances, 2) active pool - a set of unlabeled training instances and 3) testing set - data to test classification accuracy. These frameworks iteratively train a classifier by querying for labels on subsets of data from the active pool that are expected to improve classification accuracy.

A variety of selection criteria have proven to be successful in the visual domain. Holub et al. [5] select samples that minimize the expected entropy of the active pool, whereas Jain and Kapoor [6] calculate entropy from the probabilistic output of a variant of k-nearest neighbor classification to identify the most uncertain samples. Other uncertainty sampling techniques include using the posterior mean and variance from Gaussian processes classification [8], and the use of probabilistic output from a multi-class SVM to calculate the probabilistic difference between the best (most likely) and second best classes (BvSB) [7]. Li and Guo [11] combine uncertainty sampling with information density to avoid selecting uncertain samples in sparser areas of feature space which may contain outliers that are less likely to help classify more normal or denser distributions of data.

Although these frameworks yield high classification accuracy with fewer training samples, many of these frameworks still require the number of classes to be known in advance, and need a labeled seed set to prepare their multi-class classifiers. This a priori knowledge limits the frameworks to closed-world problems. Joshi et al. [7] attempt to exclude their BvSB framework from these assumptions. Although the approach requires a seed set, the seed set is chosen randomly without regard to class, meaning the number of classes does not need to be known in advance.

2.2. Unsupervised Concept Discovery

As with active learning, unsupervised visual concept discovery works with unlabeled data, but instead of finding a subset of samples to label, the goal is to partition the data by visual concept. A perfect partition would result in every group containing data from the same concept, and no two groups representing the same concept. This perfect one-to-one mapping produces groups to be labeled, allowing multiple images to be labeled simultaneously.

Tuytelaars et al. [17] provide a general overview of common unsupervised visual concept discovery techniques. Most techniques use either latent topic models [12, 16], which are adapted from the text domain to perform visual categorization, or clustering [1, 9] to create groups of visual concepts.

Unsupervised visual concept discovery algorithms are often evaluated by purity of the learned groups. This evaluation measure becomes particularly important when discussing the labeling process since only a single visual concept label is provided for an entire group of images. Unfortunately, results in the literature indicate that group purity is far from perfect. In fact, active clustering approaches [3, 18] are emerging to help partition visual data. These techniques iteratively refine the feature representation using constraint feedback from an annotator in order to get closer to the ideal one-to-one mapping that most unsupervised visual concept discovery algorithms are searching for.

Beyond the lack of sufficient partitions, many algorithms still assume the number of concepts are known in advance, even though unsupervised concept discovery presents itself as a completely bottom-up learning approach. Further, even when assuming the number of underlying visual concepts is known, not all ground truth visual concepts dominate one of the learned groups. This relates to the insufficient group purity, but also means that some concepts go undiscovered and that only easy concepts are learned.

3. Selective Guidance Approach

Selective Guidance (SG) is designed to generalize the task of visual concept discovery to any open-world data set. The algorithm makes no assumptions regarding the total number of instances, the number and types of visual concepts, or the underlying distribution of instances per class in the data set. Thus, SG does not require parameter tuning, but instead uses unsupervised learning to discover coherent groups of visual concepts that can quickly be labeled with minimal human effort.

SG begins by hierarchically clustering data, creating a cluster space that contains many groups of potential visual concepts. Discovery and labeling of these visual concepts is done iteratively. At each iteration, clusters are evaluated based on their expected information gain. Two estimates of information gain are used: 1) exploitation of the unlabeled data and 2) exploration of the visual concepts in the unlabeled data. The cluster that is expected to provide the most information gain is selected for labeling. Details of this algorithm are discussed in the remainder of this section.

3.1. Unsupervised Learning of Visual Concepts

Clustering lends itself well to unsupervised visual concept discovery because it identifies recurring patterns within data which can be used as indicators of interesting concepts. The lack of generalization to open-world data sets and poor partitioning results, however, leads us to believe
that typical spectral clustering or flat partitional clustering approaches (as in [17, 1, 10]) are not well suited for SG.

Instead, SG uses hierarchical clustering to create a hierarchy of \( n \) groups, \( \mathcal{H} = \{c_1, c_2, \ldots, c_n\} \), from the set of \( n \) training samples, \( \mathcal{T} = \{x_1, x_2, \ldots, x_n\} \). Each \( c_i \) contains at least two training samples, meaning \( m \approx n - 1 \). The two sample constraint is introduced since SG is also intended to be a technique that does not rely on instance-based labeling. By maintaining a hierarchy, the number of clusters need not be defined and the constraint of one-to-one mapping between groups and visual concepts is relaxed. This relaxation allows visual concepts to group at different locations of the hierarchy, giving smaller and more difficult classes a better shot at grouping coherently and being discovered.

Within SG, a labeling query consists of assigning a single visual concept to a cluster. Labeling the entirety of \( \mathcal{H} \) is roughly equivalent to instance-based labeling, and also redundant. Once a cluster \( c_i \) has been given a label, its descendants can inherit that label since they are subsets of \( c_i \). Thus, the task of SG is to identify a small set of labeling queries within \( \mathcal{H} \) that quickly collect meaningful information about \( \mathcal{T} \).

Queries are designed to be easy in terms of cognitive load. If a cluster is queried that represents multiple object classes, the annotator simply provides the label “mixed”, and if the cluster represents a single visual concept, i.e., it is pure, the label of that concept is provided. Note that providing a “mixed” label does not provide meaningful information that can be used to classify test data. Thus, selecting pure clusters to query is an important goal of SG.

### 3.2. Cluster Selection

Cluster selection is done iteratively, and at each iteration, clusters in \( \mathcal{H} \) exist in either the purely labeled set \( \mathcal{L} \), the mixed labeled set \( \mathcal{M} \) or the unlabeled set \( \mathcal{U} \). Selectivity is similar to active learning in the sense that \( \mathcal{U} \) is the equivalent of an active pool, and samples belonging to clusters in \( \mathcal{L} \) can be used as training data for a classifier. At each iteration, a selectivity score based on the expected information gain is given to all clusters in \( \mathcal{U} \), and the cluster with the maximum score is selected to be labeled. Two selectivity scores, one focusing on exploitation and the other on exploration are discussed later in this section. First, however, we discuss our approach to estimating cluster purity as part of estimating the exploitation selectivity score.

#### 3.2.1 Evaluating Cluster Purity

Our approach to estimating cluster purity is predicated on the belief that data near each other in feature space are more likely to represent the same visual concept than data further away in feature space, where near and far depend on the relative density of the data. Thus, clusters in \( \mathcal{H} \) that contain samples that are approximate nearest neighbors in feature space are likely to be pure. We derive a purity measure from the Proximity Forest data structure [14] which was designed to do fast approximate nearest neighbor (ANN) lookups in general metric spaces. Although any ANN algorithm would suffice for image vector data, the Proximity Forest was shown to return more accurate querying results than other common ANN algorithms such as randomized forests of kd-trees and hierarchical k-means [14]. Further, the Proximity Forest allows SG to later generalize to a broader range of data representations (e.g., videos represented as points on Grassmann manifolds [13]).

The Proximity Forest is a set of \( T \) randomized metric trees where each tree partitions data into \( \tau \) sized leaf nodes, and each leaf node encodes an approximate nearest neighborhood. Thus, data that coexist in leaf nodes in multiple metric trees are likely to be samples from the same visual concept. Treating all \( x_i \) as nodes in a graph, a weighted edge between \( x_i \) and \( x_j \) defines how connected the points are in the forest. The weight represents the number of times the data points coexist in the leaf nodes of the forest. This connectivity is extended to clusters and called the Proximity Forest Connectivity (PFC) measure.

Formally, PFC is based on the calculation of weighted edges (i.e., pair-wise connectivities) between points \( x_i \) and \( x_j \), defined as

\[
w(x_i, x_j) = \sum_{k=1}^{T} \text{common\_leaf}(x_i, x_j)
\]

where the function \( \text{common\_leaf} \) finds the leaf nodes in tree \( k \) that \( x_i \) and \( x_j \) belong to, and returns \( 1 \) if the leaf nodes are the same, and \( 0 \) otherwise. The average edge weight from \( x_i \in c_i \) to all \( x_j \in c_i \) defines how connected a sample is to its cluster:

\[
c(x_i) = \frac{1}{|c_i|} \sum_{x_j \in c_i} w(x_i, x_j)
\]

Finally, since PFC estimates purity of clusters, the connectivity of \( c_i \) is the average of each of its data point’s connectivity scores which is formally defined as:

\[
pfc(c_i) = \frac{1}{|c_i|} \sum_{x_j \in c_i} c(x_i)
\]

Higher PFC scores suggests a greater likelihood of being pure. PFC is computed without supervision since the only information being used is the relative distances between samples.

Unfortunately, PFC is not size invariant. The connectivity sum in Equation 2 is dependent on the value of \( \tau \), meaning \( x_i \) can maximally be connected to \( \tau - 1 \) other data.
points. However, the connectivity for point \( x_i \) is normalized by the size of its cluster \( c_i \). This means smaller clusters with the same connectivity as larger clusters will receive higher PFC scores, causing clusters near the leaves of the hierarchy to be favored when evaluating purity. Although this favoring is not inaccurate, it provides little benefit in terms of minimizing the labeling overhead. Further, PFC does not provide a probability of a cluster being pure, but instead a relative score. In the next section, details of how SG uses PFC to create a probability of purity for the exploitation score are discussed.

### 3.2.2 Exploitation

Exploitation favors collecting a large number of labeled samples quickly. Thus, cluster purity and cluster size are important factors in information gain based on exploitation. The SG exploitation score for a cluster \( c \) is defined as

\[
\text{exploit}(c) = p_c \times l_c
\]

where \( p_c \) is the probability that \( c \) is pure and \( l_c \) is a count of unlabeled leaf node descendants of \( c \). Note that \( l_c \) is not the size of \( c \) since some samples may already be labeled if a descendant of \( c \) was selected for labeling in a previous iteration.

The value of \( p_c \) is calculated by iteratively modeling PFC scores and cluster sizes from clusters that exist in \( \mathcal{L} \) and \( \mathcal{M} \), PFC score and size. Recall that PFC is not size invariant, thus, adding cluster size to the model helps ensure that selectivity of clusters is not limited to the bottom of the hierarchy. As the number labeled clusters grows, more information is available to predict what ranges of PFC scores and sizes are more likely to be pure. This information is modeled using a 10x10 uniform grid of Gaussian radial basis functions (RBFs). One axis of the grid represents the range of PFC scores while the other represents the range of cluster sizes. Each axis is normalized to \([0, 1.0]\), allowing for an even distribution of grid point centers along these axes.

The RBF grid is modeled online as labeling queries are processed, and each grid point is modeled as the average weighted purity of the current labeled clusters. That is, after \( t \) labeling queries, grid point \( g_i \) has a modeled purity value of

\[
p(g_i) = \frac{1}{t} \sum_{i=0}^{t} \phi(r_i) \times v_i
\]

where \( r_i \) is the distance between the grid point center and the cluster queried at iteration \( i \). \( v_i = 0 \) if the cluster was labeled “mixed” or 1 if it was given a label corresponding to a visual concept, and \( \phi(r_i) \) is the Gaussian RBF formally defined as \( \exp(-r_i/\sigma)^2 \). For all experiments in this paper, \( \sigma = 0.1 \). Clusters closer to the center of \( g_i \) will be weighted more heavily than clusters further the center of \( g_i \).

The value of \( p_c \) is calculated from the RBF grid as

\[
p_c = \phi(r_c) \times p(g_i)
\]

where \( g_i \) is selected as the grid point that minimizes \( r_c \). The exploitation score describes the expected number of samples that will receive labels if a visual concept label is provided. In this selectivity approach, the focus is on how to gain information about as many samples in as few of queries as possible. Focusing solely on exploitation, however, favors the discovery of visual concepts that are easy to group and that dominate the data set which may not result in a large amount of exploration of the data set.

### 3.2.3 Exploration

Exploration focuses on how to discovery different visual concepts quickly. Exploration is modeled in SG with the assumption that often, different visual concepts will be located in different areas of feature space. Thus, when selecting a cluster from \( \mathcal{U} \) to be labeled, it should be far away from the clusters that already exist in \( \mathcal{L} \) in order to try and identify a new visual concept.

The exploration selectivity score is based on a distance value and defined as

\[
\text{explore}(c) = \min_{\forall c_i \in \mathcal{L}} d(c_i, c)
\]

where \( d \) is a distance function measured as Euclidean distance between two cluster centroids. After two labeling query iterations, each cluster will have multiple distances between the clusters in \( \mathcal{L} \). The minimum distance from \( c \) to any cluster in \( \mathcal{L} \) is used since it represents the difference between \( c \) and its most similar labeled neighbor. The cluster in \( \mathcal{U} \) with the maximum exploration score represents the cluster that is most dissimilar to its nearest neighbor, and is selected to be labeled.

Note that for both selectivity scores, SG does not rely on retraining a particular classifier after each labeling query. The only supervised modeling that is done comes at the level of cluster purity within the RBF grid for the exploitation score.

### 4. Experiments

This section includes experimental results that compares the classification accuracy yielded after collecting training data via SG and from the BvSB active learning framework of Joshi et al. [7]. Recall that this framework also takes a completely unsupervised approach to collecting labeled training data, and therefore requires few caveats to make a direct comparison. This active learning framework does however, require a seed set. Although the seed set is not labeled during active learning, an annotator must still provide labels at some point in order to have labeled data. Thus, it is
assumed that the total number of labeling queries answered during the BvSB approach is $|S| + m \times r$ where $S$ is the seed set, $m$ is the number of active learning rounds and $r$ is the number of samples selected per round.

We present two classifier results using SG. Using clusters in $L$, the label of each cluster is transferred to its samples. These labeled samples are then used in a Nearest Neighbor (NN) classifier, and also used to train a Support Vector Machine (SVM). We use the same SVM kernels that are reported by Joshi et al.

Further, a deeper evaluation of the the exploitation ($SG_{exploit}$) and exploration ($SG_{explore}$) selection criteria is given in this section. This evaluation and comparison not only focuses on the classification accuracy but also on how quickly classes are discovered and samples are labeled.

### 4.1. Data Sets

Four benchmark data sets are used for evaluation. The first three data sets and experimental protocols are replicated from the work of Joshi et al. The first two are UCI data sets, Pendigits and Letter. The Pendigits data set contains hand written digits from 44 writers, and the Letter data set contains images of English capital letters in 20 different fonts. The third is the 13-Scenes [2] data set which contains images from 13 categories of natural scenes. For this data set, GIST [15] features are used to represent each image as in Joshi et al. The exact training and testing partition is not reported by Joshi et al. for the 13-Scenes data set. We report the SG results using a 2,500 training and 500 testing partition, but found similar trends across several other partitions.

These first three data sets contain visual concepts that have roughly the same image distribution per class. Open-world data sets, however, may not necessarily have an even distribution of concepts. Thus, the fourth data set used for evaluation contains classes with an uneven distribution of images. This data set consists of a commonly used subset of 20 classes [17] from the Caltech-256 [4] data set. The 20 classes range from 83 to 800 images. Images are represented using the publicly available 1,000 dimensional Bag of Words feature vectors from Tuytelaars et al. [17].

Table 1 provides experimental details of all four data sets. All experiment results are averaged over 20 trials.

### 4.2. Selective Guidance vs Active Learning

High classification accuracy with limited labeling queries is the ultimate goal for both SG and active learning. Figure 1 shows the classification accuracy per labeling query for the first three data sets using both SG selection criteria and both SG classification methods. Common trends can be seen among the results of all three data sets. First, all SG methods outperform the active learning framework early on in the labeling queries. This suggests that given a time constraint where an annotator is only able to provide a limited number of labeling queries, SG would likely collect a more diverse set of informative data faster than the active learning framework.

Second, at least one of the classification methods used with SG, regardless of the selection criteria, outperforms the active learning approach. The active learning framework does converge to the SG results on the Pendigits and Letter data sets, but on the whole, SG classifies at least as well, if not better than the approach of Joshi et al. The smallest performance gap is seen in Pendigits, but this is the least challenging of all the data sets because with only 350 labeling queries (less than 7% of the entire training data set), very high classification accuracy is achieved for both SG and active learning.

The performance gap seen on the 13-Scenes data set is significantly larger than that seen on the UCI data sets. The active learning framework never converges to the performance of SG even after 1,000 labeling queries. This is likely due to the value of $r$ that Joshi et al. used for this data set. They set $r = 20$ to claim that even though potentially redundant data are being labeled at each active learning round, they still achieve improved results over random selection. This may be true, but because SG first learns to group some of these redundant data, they can be labeled simultaneously with a single labeling query.

Further comparisons between $SG_{exploit}$ and $SG_{explore}$ can be made. In terms of classification, $SG_{exploit}$ and $SG_{explore}$ perform similarly and provides little insight into how the performance of the two selection criteria differ. To understand the trade-off between the two criteria, we look at the number of discoveries and the number of individual samples that receive labels per labeling query.

Figure 2 shows the number of discovered classes per query for each SG method on the three data sets. As is the intended goal, $SG_{explore}$ discovers visual concepts more quickly, which provides a classification advantage early in the querying process since the only way to correctly classify a test sample is to first discover and label its visual concept. This may have contributed to the early performance jump that $SG_{explore}$ achieved in the Pendigits data set. After about 100 queries, the performances converge (and then later cross), which is about the time that $SG_{exploit}$ has also discovered all 10 visual concepts in the data set.

Figure 3, however, shows there is also a significant difference in terms of the number of samples that receive labels throughout the querying process. $SG_{exploit}$ clearly outperforms $SG_{explore}$, which likely causes the cross in the classification accuracy in the Pendigits data set since more training data is available for classification. Notice however, that both SG methods assign class labels to individual samples at a faster rate than the active learning instance-based labeling approach.
<table>
<thead>
<tr>
<th>Data Set</th>
<th>Total Instances</th>
<th>Training Size</th>
<th>Testing Size</th>
<th>SVM Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCI-Pendigits</td>
<td>10,992</td>
<td>5,100</td>
<td>2,000</td>
<td>Linear</td>
</tr>
<tr>
<td>UCI-Letters</td>
<td>20,000</td>
<td>7,100</td>
<td>5,000</td>
<td>RBF</td>
</tr>
<tr>
<td>13-Scenes</td>
<td>3,859</td>
<td>2,500</td>
<td>500</td>
<td>Linear</td>
</tr>
<tr>
<td>Caltech256 - 20 selected</td>
<td>3,756</td>
<td>1,879</td>
<td>1,887</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 1: Experiment details for benchmark data sets.

Figure 1: Classification accuracy per labeling query on data sets (a) Pendigits, (b) Letter and (c) 13-Scenes.

Results on the first three data sets show that SG performs at least as well if not better than the unsupervised active learning framework of Joshi et al. Further, a selection criteria based on exploitation results in a much larger number of samples labeled per labeling query whereas a selection criteria based on exploration results in the visual concepts being discovered earlier in the labeling queries. These trade-offs, however, do not seem to affect the classification performance in such a way that would indicate one selection criteria is clearly better than the other.

4.3. Uneven Distribution of Images Per Object Class

Since the Caltech-256 subset has an uneven distribution of images per class, an overall classification accuracy can be misleading. Discovering only a single object class that dominates the data set may yield a reasonable overall classification accuracy when in fact the majority of interesting concepts are never being learned. Thus, in addition to the overall classification accuracy, the mean classification accuracy per class is reported.

Figure 4 shows the NN classification results for the two selection criteria. The performance gap between $SG_{\text{exploit}}$ and $SG_{\text{explore}}$ is fairly small in terms of overall classification (Figure 4a). $SG_{\text{exploit}}$ has a performance edge until just after 100 labeling queries. We see in Figure 5b that the number of labeled samples per labeling query is steepest for $SG_{\text{exploit}}$ up to this same point, indicating that clusters with a large number of samples were labeled and bumped up the classification accuracy.

However, the first 100 labeling queries is where $SG_{\text{exploit}}$ makes a huge leap in terms of discovery relative to $SG_{\text{exploit}}$, as seen in Figure 5a. Shortly after the first 100 queries, the overall classification performances of the two criteria cross, and although slight, $SG_{\text{explore}}$ does maintain better classification performance for the remaining labeling queries. Even with the uneven distribution of images per object class, the discovery of more object classes did help yield better overall classification than simply learning to classify the dominating classes.

More commonly reported for unevenly distributed data sets, however, is the mean classification accuracy per class which can be seen in Figure 4b. This comparison shows a much more significant gap between the two selection criteria. To even begin classifying certain object classes, they must first be discovered. $SG_{\text{exploit}}$ sacrifices this exploration in order to gain a large amount of knowledge in terms of quantity of samples that are labeled, causing mean classification to be compromised.

Clearly with a skewed image data set, exploration becomes particularly important. Exploration and discovery are not easy tasks though. Figure 5a show that both selection criteria have a difficult time reaching the discovery of all 20 classes after 500 labeling queries. $SG_{\text{explore}}$ nearly reaches this point after 200 queries but struggles to find the remaining one or two classes in the last 300 queries. The process of finding these rare but interesting concepts becomes particularly important in order to classify unevenly distributed data sets.
5. Conclusions and Future Work

Reducing the labeling overhead to collect training data has been addressed in a variety of ways. However, there is a common lack of generalization to open-world data sets because most approaches assume there is an a priori knowledge of what visual concepts the unlabeled data set contains. We introduced Selective Guidance (SG) as an open-world visual concept discovery approach that identifies easy to label clusters in order to collect labeled training quickly.

Using four benchmark data sets, we showed that SG is able to discover the underlying visual concepts and query about these groups of data to collect training data that yields classification accuracy at least as good as active learning frameworks. Two different selection criteria were introduced for SG. As with other forms of learning, there is a trade-off between exploitation and exploration within the selective guidance cluster selection criteria. However, for data sets that have an uneven distribution of images per class, the exploration selection method proved to outperform the exploitation method. It is not enough to learn a lot of information about the training data, but the learned information also needs to be somewhat diverse.

The exploitation criteria eventually learned diverse information but at a much slower rate than the exploration selection criteria. However, the exploitation method is able to quickly learn information about large quantities of data. These criteria have been evaluated independently, but future work will look at the weighted combination of exploitation and exploration to further improve SG.

References

Figure 4: Classification accuracy per labeling query on a subset of 20 classes from the Caltech-256 data set: (a) overall classification accuracy and (b) mean classification accuracy per class.


Figure 5: Number of classes discovered (a) and number of samples labeled (b) per labeled per query for the Caltech-256 subset.


