A Semi-Automated Labeling Approach for a Hierarchy of Image Clusters

Maggie Wigness
Colorado State University
Fort Collins, CO
mwigness@cs.colostate.edu

Bruce A. Draper
Colorado State University
Fort Collins, CO
draper@cs.colostate.edu

ABSTRACT

Many computer vision learning tasks require a set of annotated images to train from. As datasets grow very large, the human labor needed to complete the annotation process becomes very demanding. Approaches have been proposed that attempt to reduce the dataset in some fashion in order to alleviate the annotation workload, but these approaches still require the annotator to provide labels for each image in the reduced set. This paper introduces a new approach that uses hierarchical clustering to semi-automate the labeling of an image dataset. Using a small number of human annotations, the system automatically propagates labels to clusters unseen by the user. Further, this paper demonstrates how images extracted from videos can lead to further automatic labeling.

Keywords

Automatic label propagation, Hierarchical image clustering

1. INTRODUCTION

In computer vision, collecting data in the form of images or videos is a trivial task. However, the cost of obtaining annotation for this data comes at a high price that requires a significant amount of human labor. Unfortunately, data annotations are necessary to train classifiers and detectors, and performance of such systems often improves with more training data, not less. As an example, training an object detector requires a human to view a series of images and provide a label that describes the object in the image, such as *person* or *car*. The complexity of this task grows when multiple objects exist in an image because in addition to a label, a bounding box describing each object's location is often required.

Hand annotating a large set of images is a problem that often needs to be addressed in object recognition/classification problems. Research done in the active learning and unsupervised object discovery communities has led to methods that can be used to help with the burdensome task of labeling such datasets. The active learning community has proposed intelligent selection methods to build a subset of images that is representative of the entire dataset. Eliminating a fraction of the dataset causes the workload to be reduced by the same fraction. Unsupervised object discovery techniques try to divide the dataset into a small number of groups, such that each group represents a single object class. Grouping the images reduces human labor by moving from instance-based to group-based labeling.

Each method performs a reduction in the number of items that need to be labeled, but after the reduction the human annotator must label everything in the reduced set. This creates a scenario in which researchers have to choose between a burdensome annotation workload and a coarse-grained dataset. Choosing coarse granularity can result in distinctive instances being eliminated or groups becoming too large so they no longer represent a single object class.

This paper introduces a semi-automated labeling approach for images extracted from a video dataset. Images are hierarchically clustered to create a large set of fine-grained clusters. The goal of this work is not to label the entire hierarchy of clusters, but to demonstrate that there are ways to automatically propagate and infer labels to reduce the human workload. In this system, a human annotator is asked to hand label a small fraction of the clusters in the hierarchy, and the provided label information is used to propagate labels to unlabeled relatives in the hierarchy. The success of automatic label propagation depends on the intelligent selection of clusters that the annotator is asked to label. Additionally, the images are extracted from a video in which objects are tracked over a series of frames. This tracking information provides further opportunities to automatically infer labels for clusters never seen by the annotator.

This paper will be organized as follows. Section 2 provides greater details about the current methods being explored to reduce the annotation workload. Section 3 provides a general overview of the proposed system. A description of image extraction and the implications of hierarchical clustering are provided in Section 4. The iterative selection and labeling process is described in Section 5. Section 6 provides initial results that demonstrate the potential for automatic label propagation, followed by concluding remarks in Section 7.

2. BACKGROUND

Active learning techniques are applied to many data domains. Within the image domain, techniques use appearance descriptors to identify a subset of image instances that represent the diversity found within the dataset as a whole. Pool-based sampling is most closely relative to the work being done in this paper. In this technique, there exists a large pool of unlabeled image samples, and images are selected from the pool to be labeled by a human annotator. Selection is based on the calculation of information uncertainties such as expected entropy [1], class membership probabilities [2], or low-level feature probabilities [9]. Active learning pool-based sampling provides a reduction in training samples, but the reduced set must be labeled in its entirety. For very large datasets containing many object classes, this reduction may still require a significant amount of human labor effort.

Unsupervised object discovery techniques group image instances into cohesive sets, creating a reduced dataset by moving from instance-based to group-based labeling. Several approaches exist in the object discovery field which can be reviewed in the comparison of techniques provided by Tuytelaars et al [8]. However, most approaches use either a latent topic model or clustering algorithm for discovery.

Latent topic models are derived from data mining techniques used to perform topic discovery in documents, but adapted for the image domain. Words in documents become image features, and document topics become an object class. Latent topic model approaches [5, 7] differ in their object class modeling, but generally rely on extracting appearance features as a "bag of words" (e.g. extracting SIFT descriptors from interest points), and mapping the frequencies of said visual words to a learned object class topic.

A variety of clustering algorithms have been used for object discovery, but the common ground shared by most approaches [3, 8] is the focus on batch or single shot discovery. Meaning that all object classes are discovered at once, in a single iteration. Deviating from batch discovery, Lee and Grauman [4] present an iterative object discovery approach that learns object classes in order of difficulty, easiest to hardest. This iterative discovery process is similar to the iterative labeling/discovery that this paper proposes.

Most of these described object discovery approaches require a parameter k that defines the number of groups to divide the images into. More times than not, the value of k is set to represent the number of object classes in the dataset. In large datasets, this often results in the construction of clusters that are very coarse, and gives rise to low cluster purity measures (i.e. images from multiple object classes are represented by the clusters).

3. SYSTEM OVERVIEW

This work can be broken down into two major phases. The first phase includes the tasks that must be completed to establish the structure needed for the iterative annotation and label propagation. These tasks include extracting image samples from the video dataset, and hierarchically clustering the images. These pre-labeling tasks are discussed in greater detail in Section 4. The second phase represents the iterative annotation and automatic label propagation. At each iteration a cluster from the hierarchy is selected and displayed to the human annotator. The annotator provides a label describing the object class the cluster represents. The label is then used to propagate information to other unlabeled clusters. Figure 1 shows an example of a single iteration in the labeling process. Details of the labeling process are provided in Section 5.

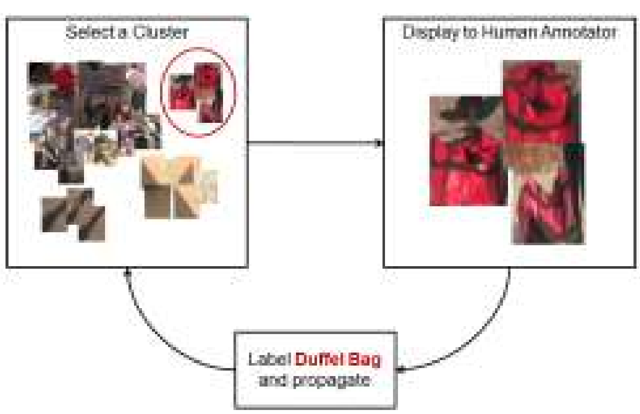

Figure 1: Flow of events for each iteration in the labeling process.

This paper is intended to introduce a pilot version of the described semi-automated labeling system. The system is designed to maximize the number of leaf nodes in the hierarchy that receive class labels, while minimizing the human annotator workload. Attempting to hand label the hierarchy in its entirety is unrealistic given the fine-grained structure that has been defined. However, this work will demonstrate that the pilot system can take human provided cluster labels and propagate or infer labels for a much larger percentage of the hierarchy.

4. PRE-LABELING TASKS

4.1 Image Extraction

The image samples used in this work are extracted from regions tracked over time in a video. Objects of interest are defined as anything having motion. A low level vision process identifies the location of moving objects and issues a tracker to follow them over time. Figure 2 shows a single frame from a video in which a track (illustrated as a blue rectangle) is instantiated to follow a person who is walking.

Figure 2: A track (blue rectangle) bounding a person.

The set of images to be labeled are extracted from the bounded regions defined by tracks at various frames. For instance, if a track exists in a video for frames 300 to 600, the low level vision system may extract four samples at frames 307, 425, 510, and 555. The rate of sample extraction is based on significant appearance changes seen in the foreground pixels of the bounded region. Thus, the number of image samples extracted from a track will depend on track length, and the amount of motion or pose variation an object undergoes.

Figure 3: Extracted image samples from the same track. Images are from frames 61, 144, and 183 (left to right).

Using a video dataset to collect images presents a unique opportunity for automatic label propagation (details provided in Section 5.3). Assuming the tracker maintains focus on the same object over time, image samples from the same track should share the same label. Figure 3 shows three images extracted from the same track, and although the appearance is different between each sample, they all belong to the same *person* object class.

Once an image has been extracted from a track, a set of features are calculated for the image. The feature vector used in this work has 164 dimensions comprised of a pyramid of Histogram of Oriented Gradients, image moments, color descriptors, and a set of descriptors pertaining to foreground pixels and image size.

4.2 Hierarchical Clustering

Clustering approaches fall into either the partitional or hierarchical category. Partitional clustering algorithms divide the dataset into *k* groups, and typically the value of *k* is selected before clustering begins and is passed to the algorithm as a parameter. Selecting the value of *k* is not a straightforward process. A large value of *k* makes labeling and inspection labor-intensive, and often the clusters are too specific and do not generalize well. A small value of *k* often results in clusters of large sizes that do not represent a single object class because of many outlier instances.

Partitional clustering is common, and when working with images the value of *k* is often chosen to reflect the number of object classes represented in the dataset. However, this implies that information about the number of object classes is known a priori which is not always the case. Even if that information is known, clustering algorithms are designed to identify similarities and group those instances together. Ideally, the similarity that the clustering algorithm identifies represents the object class, but this cannot be guaranteed. Instances from the same object class may look different because of pose variability which was seen in Figure 3 (e.g., a *person* may be standing or bending), or changes in perspective (e.g., a *car* can be viewed from the side or from behind).

The difficulties involved with the selection of *k* suggest that it is an understated problem. A fine-grain value of *k* may be best suited for some object classes while a coarser-grained value of *k* is ideal for other object classes. Most image datasets contain a set of object classes that represent a large spectrum of ideal *k* values. For these reasons, this system does not force a one-to-one mapping of clusters and object classes, and avoids the difficulty of selecting a value of *k* by using a hierarchical clustering algorithm.

Hierarchical clustering continually splits the dataset into smaller subsets (top-down), or continually merges instances together until a single group is formed (bottom-up). The resulting structure is a hierarchy of clusters with a root that contains all image instances, leaf nodes that contain a single image instance, and many internal nodes that contain a subset of all images. Hierarchical clustering allows for each object class within the dataset to break down (or build up) into groups at its own pace. The number of groups, *k*, is determined at the time of construction.

An optimized agglomerative (bottom-up) clustering implementation [6] that uses Ward's linkage criteria and Euclidean distance metric is used to create the hierarchical structure for this work. To reduce the number of clusters in the structure, a minimum size threshold of 20 is placed on all clusters. This effectively removes the bottom portion of the hierarchy that contains small clusters that are unlikely to generalize well. The minimum cluster size was selected based on the amount of reduction the threshold would provide to the hierarchy. Different threshold values are not investigated in this study, but it is noted that the selected value of 20 may be too coarse for object classes with very few samples. Future work will look at automatically determining when the size threshold should be varied.

4.2.1 Cluster Types

The hierarchical structure is made up of two types of clusters defined as follows:

- **pure** – contains images from a single object class
- **mixed** – contains images from multiple object classes

The most obvious example of a *mixed* cluster (assuming a multi-class dataset is being used) is the root of the hierarchy that contains all image samples. The trivial example of a *pure* cluster would be the original leaf nodes that contain only a single image sample. This paper uses the phrase "pure cluster" to refer to any cluster that receives an object class label, either from the annotator directly or via label propagation. Figure 4 (a blown up subpart of Figure 1) shows four pure clusters that represent the object classes *person*, *duffel bag*, *ground*, and *shadow*. Any cluster that is not pure will be identified with the label *mixed*.

Figure 4: Examples of pure clusters: *person, duffel bag, ground, and shadow.*

4.2.2 Label Propagation

The hierarchical structure naturally defines an important type of label propagation based on the fact that descendant clusters contain a subset of images found within its ancestors. Thus, if a pure cluster is given an object label, each of its descendants can automatically inherit the same object label. Conversely, if a cluster is given a mixed label, each of its ancestors can automatically inherit the mixed label since they will contain all the image instances that have been identified as a mixed set.

The described label propagation suggests that the annotation workload can be greatly reduced if pure clusters with many descendants are selected for the user to hand label. The modeling and selection process used to exploit this label propagation is described in the next section.

5. ITERATIVE LABELING

Thus far, the stress has been on minimizing the human workload by maximizing the amount of automatic label propagation. However, it should also be noted that the human workload is minimal in terms of cognitive load. A cluster of images is displayed to the user, and the only decision that has to be made is whether or not the cluster is pure or mixed. If the cluster is pure, naming the object class should be trivial. The low cognitive load of this labeling task allows the user to complete many iterations in a small amount of time.

Although each iteration can be completed quickly, annotators may not have the time (or desire) to spend several days answering easy questions to label a large dataset. The next sections discuss the modeling and selection that happens within the system in order to avoid thousands upon thousands of labeling iterations.

5.1 Cluster Modeling

Cluster purity plays an important role in the selection process because pure clusters provide informative label propagation. Mixed clusters on the other hand do not provide any concrete information about what object classes the dataset contains. This system models purity using the number of images that a cluster contains and variance. Cluster variance is defined as the average squared distance of each image sample to the centroid of the cluster. More formally the variance of a cluster, v_c, that contains n image samples is defined as

$$v_c = \frac{1}{n}\sum_{i=0}^{n}(s_i - d_c)^2$$

where s_i is an image sample, and d_c is the cluster centroid, both of which are 164 dimensional feature vectors.

Average variance is selected as a purity descriptor because it can describe the compactness of a cluster. Image samples that are all near the centroid of the cluster indicate self-similarity which can be a good indicator of cluster purity. Clusters that are less compact may have outlier instances that do not resemble the majority of image instances within the cluster.

The number of images in the cluster can provide context for the variance. A cluster with a large variance and a small number of images is more likely to be mixed than a cluster with the same large variance that has many image samples. Thus, the system attempts to model cluster purity using variance and image count.

The mechanism used for purity modeling is a uniform 10x10 grid of Gaussian radial basis functions (RBFs). The grid axes represent the range of values for image count and variance found in the hierarchy. These ranges are normalized to values between 0 and 1, and the value of σ is set to 0.1. Each axis is uniformly split into 10 sections and the centers of the RBFs follow this uniformity.

Prior to settling on the use of a grid of RBFs, simple one-dimensional Gaussian models were explored. Using only variance, one model was constructed for pure clusters and a second model for mixed clusters. However, different object classes have different typical variance ranges. The dataset used in this work has an uneven distribution of object classes, meaning the pure Gaussian model was dominated by a single object class, or a single variance range. The grid of Gaussian RBFs is better fit to handle distribution skews.

The grid models ranges of variances and image counts that clusters may fall into. Each grid point keeps track of what types of clusters (pure or mixed) are falling into its specified range so that when an unlabeled cluster falls into its range, the RBF can determine whether it is likely to be pure or mixed. This is done by maintaining an average weighted score for the t clusters that have been labeled and fall into the range of grid point g. The average weighted score is defined as

$$avg_g = \frac{1}{t}\sum_{i=0}^{t}\phi(r_i) * l_i$$

where r_i is the distance from the cluster to the RBF center, l_i is 0 for a mixed cluster or 1 for a pure cluster, and $\phi(r_i)$ is the Gaussian RBF formally defined as

$$\phi(r) = e^{-(r/\sigma)^2}.$$

The RBF grid is updated iteratively with the labeling process. As the annotator provides a label for a cluster, its descriptors and the descriptors of any clusters that received labels via propagation, are added to the appropriate locations in the grid.

5.2 Cluster Selection

At each iteration, there is a set of clusters that have yet to be labeled. Cluster selection is based on which unlabeled cluster is expected to provide the best automatic label propagation. As mentioned earlier, maximal label propagation is dependent on the likelihood that a cluster is pure, and number of unlabeled descendants that a cluster has.

The probability that a cluster is pure, p_c, is defined as the weighted best fit in the RBF grid. Each cluster falls into a grid point g. The cluster is fit to g using the Gaussian RBF and weighted by the average score from previously seen labeled clusters. Thus, the likelihood of a cluster being pure is

$$p_c = \phi(r_c) * avg_g.$$

This likelihood is weighted by the number of unlabeled descendant leaf nodes of the cluster, d_c, resulting in an expected propagation score of

$$s_c = p_c * d_c.$$

The cluster with the maximum expected propagation score is issued to the annotator for labeling.

5.3 Post-Labeling Inference

The primary push for automatic labeling comes from the propagation of labels to descendants and ancestors discussed in Section 4.2.2, but a second type of automatic labeling is achieved after an annotator has decided to stop labeling. This automatic labeling comes through an inference that can only be made because the image samples are extracted from a video dataset.

If a sample belongs to a cluster given the label *person*, then it can be inferred that an unlabeled cluster that contains a different image sample from the same tracked region is also likely to represent the object class *person*. Again, this is because multiple images are extracted from the same tracked region, and a track is designed to follow the same object over time.

Track inferences are made post-labeling. Every labeled cluster has a set of tracks associated with it. Each of those tracks will inherit the label of the cluster. This produces a set of labeled tracks, and in some cases, a track may have been labeled many times if multiple images extracted from it belong to different labeled clusters. This set of labeled tracks is used as evidence to support label inferences for the remaining unlabeled clusters.

To determine the evidence for label inferences, the images of a cluster use their track reference to perform a lookup in the set of labeled tracks. If the image's track reference is in the set of labeled tracks, then it votes for the label it found during the lookup. If the image's track reference is not found in the set of labeled tracks it votes for an *unknown* label. If enough evidence for a particular label is seen during voting, the unlabeled cluster can automatically be labeled. Figure 5 shows an example of an unlabeled cluster with seven instances illustrated as their video and track references. Three of the instances perform a successful lookup in the labeled track set. The label *car* could be inferred for the cluster given by the evidence score of 0.428 which is simply defined as the ratio of successful lookups to the total number of image instances.

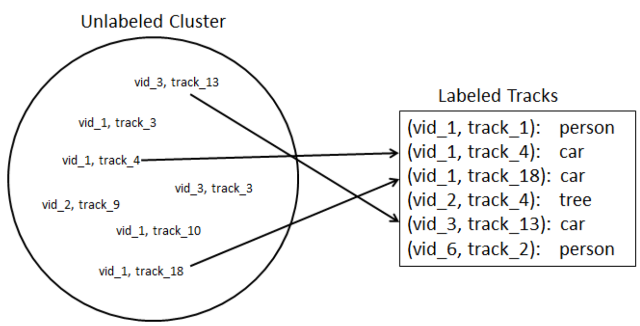

Figure 5: Track inference example. There is evidence that the unlabeled cluster contains *car* images.

6. LABELING RESULTS

This section demonstrates the potential of an automatic label propagation system. Specifically, the results show the consequences of automatic labeling during 500 iterations of the previously described labeling process. This demonstration is completed using a video dataset containing 191 videos. The videos are of people performing various actions, some of which include walking, exchanging objects, riding bikes, or carrying boxes. The dataset was distributed by DARPA for the Mind's Eye project, and will hopefully be publicly released in the near future.

Table 1: Basic information about the dataset and the hierarchical structure produced from the dataset.

Number of Tracks in Dataset	151,394
Number of Images Extracted From Tracks	274,003
Total Clusters in Hierarchy	31,369
Total Leaf Nodes in Hierarchy	7,777

Table 1 provides some basic structural information about the dataset and the hierarchy of images. There are a large number of extracted images which would make hand labeling extremely labor intensive. Hierarchically clustering the instances produces a set of items to label that is more than 35 times smaller than the set of individual images, assuming the annotator would provide labels only for leaf nodes.

6.1 Label Propagation

Results in this section demonstrate the automatic label propagation to cluster relatives during the iterative labeling process. Table 2 provides details on the number of clusters in the hierarchy that received labels after 500 hand annotations. The number of hand annotations accounts for a very small portion of the entire hierarchy, less than 2%, but the results demonstrate that there is almost an order of magnitude of propagation that is occurring. Of the 4,813 clusters that received labels, 1,421 of them are leaf nodes. Thus, after 500 hand annotation, 18.3% of all leaf nodes have been labeled.

Table 2: Propagation results for all clusters in the hierarchy.

Hand Labeled	Propagated	Total
500	4,313	4,813

6.2 Purity Modeling

The benefit of label propagation to leaf nodes only applies when the annotator provides a pure object class label for the selected cluster because a mixed label only propagates up the hierarchy. Unfortunately, only 199 of the 500 clusters selection for hand labeling were given pure labels. Thus, for the majority of iterations, label propagations were made to ancestors and the number of labeled leaf nodes remained the same.

These results suggest that variance and image count are not strong enough features to model cluster purity on their own. However, recall that the selection score is based on the likelihood of being pure and the number of unlabeled leaf node descendants. One could imagine a cluster with high probability of being pure and few leaf nodes. It is likely that the cluster's parent has twice as many unlabeled leaf node descendants (assuming the hierarchy resembles a balanced binary tree), and as long as the parent cluster has a likelihood of being pure that is no less than half of its child's, the parent cluster will produce a higher selection score.

This suggests that the selection score calculation allows potential label propagation to dominate the purity probability. This domination factor has good intentions, but if it dominates incorrectly then the annotator's efforts are essentially being wasted. Future work will investigate other purity measures to improve modeling, including reoccurring image connectivity in non-deterministic or subspace clusterings. A set of images that often cluster together could indicate a pure coherent grouping. Improving both the modeling descriptors and the dominance issue described could lead to more pure cluster selections.

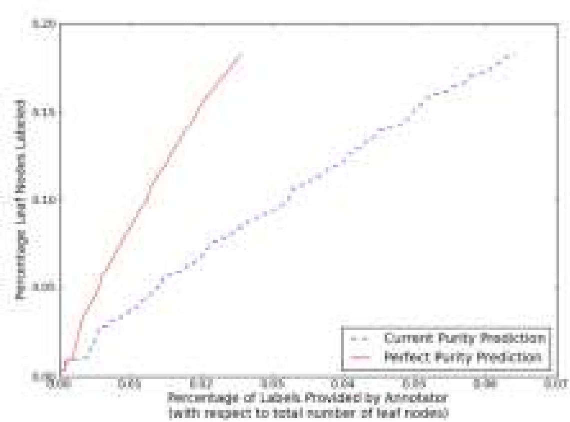

Figure 6: Comparison of purity models using a hand labeled to total labeled ratio.

To illustrate the importance of pure cluster selection, a simulated version of the labeling process is done using only the 199 pure clusters that the original selection process chose. Figure 6 plots the hand labeled to total labeled ratio for the original selection model, and the "perfect" selection model. These ratios are based on the number of leaf nodes labeled. This figure shows that with perfect purity selection, the system is much closer to reaching an order of magnitude of propagation to the leaf nodes. It is unknown if this trend would continue with further labeling, but it demonstrates the potential for effective automatic label propagation.

6.3 Track-Based Inference

Track-based inferences are done post-labeling for the remaining unlabeled leaf nodes. After 500 labeling iterations, 33,421 tracks have implicitly received labels. Of this set, 4,020 tracks were labeled more than once (i.e. they were found in multiple clusters that were labeled). However, only 4 of them had disagreeing labels which may be a good indicator that the assumption made about tracks not deviating from the object they are called to follow is true more times than not.

Track inferences are performed on the remaining 6,356 unlabeled leaf nodes. 5,189 of these clusters found at least one track match in the set of labeled tracks used for lookup. This means that after only 500 hand annotations (less than 2% of entire hierarchy, less than 7% of total leaf nodes), 85% of leaf nodes could potentially be labeled. Figure 7 shows the distribution of label evidence ranges for all 6,356 unlabeled clusters. The right half of the figure shows a small percentage of clusters that received a large amount of evidence. Less than 10% of the clusters matched at least half of their track references. This means that many of the inferences would be low confidence guesses.

There was no empirical study pertaining to how much evidence is enough to infer a correct label so this remains an open question. However, track inferences are being made across leaf nodes, none of which contain more than 40 images. These clusters are small and are the most likely clusters in the hierarchy to be pure. Thus, if any low evidence inference is going to be made, it is best to be made on the set of leaf nodes. Additionally, 4,370 of the leaf nodes had multiple track lookup successes, but only 23 (less than 1%) of those clusters received conflicting evidence. This is by no means sufficient proof that the leaf nodes in the hierarchy are always pure and therefore only need a single track lookup success to be inferred correctly. However, it does support the belief that leaf nodes are the safest clusters to perform track inference across.

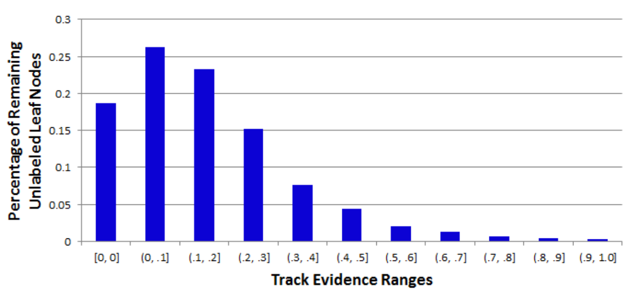

Figure 7: Distribution of evidence seen in the remaining unlabeled leaf nodes.

6.4 Object Discovery

The focus of the results (and this paper in general) has been on automatic label propagation. However, a consequence of labeling is the discovery of different object classes. In this small experiment of 500 iterations of labeling, the system was able to discover 9 unique object classes. The dataset used in this experiment is truly unlabeled meaning there is no ground truth information available. Without ground truth it is difficult to measure this value and determine how well the system is suited for object discovery. A quantitative analysis of the number of object classes discovered will be looked at in future work.

7. CONCLUSION

The rapid growth of technology makes it possible to capture a high volume of data with very little effort. However, collection of large datasets can require a significant amount of human labor to annotate the collection with ground truth labels. This work has shown that using a hierarchical clustering algorithm to group a set of data provides a unique structure that provides automatic label propagation for each hand labeled annotation a user provides. Additionally, images extracted from tracked regions of a video possess an important attribute that provides label evidence for unseen clusters. It was demonstrated that by exploiting these two characteristics, a large portion of clusters could be automatically labeled without human intervention. As datasets continue to grow, automatic labeling techniques will become a necessity to help alleviate the annotation workload.

8. REFERENCES

[1] Holub, A. D., Perona, P., and Burl, M. C. Entropy-Based Active Learning for Object Recognition. In *Online Learning for Classification Workshop*, pages 1-8, 2008.

[2] Joshi, A. J., Porikli, F. M., and Papanikolopoulos, N. P. Multi-class Active Learning for Image Classification. In *CVPR*, pages 2372-2379, 2009.

[3] Lee, Y. J., and Grauman, K. Object-Graphs for Context-Aware Category Discovery. In *CVPR*, pages 1-8. IEEE, 2010.

[4] Lee, Y. J., and Grauman, K. Learning the Easy Things First: Self-Paced Visual Category Discovery. In *CVPR,* 2011.

[5] Liu, D., and Chen, T. Unsupervised Image Categorization and Object Localization using Topic Models and Correspondences between Images. In *ICCV*, 2007.

[6] Mullner, D. fastcluster. Fast Hierarchical Clustering Routines for R and Python.

[7] Russell, B. C., Freeman, W. T., Efros, A. A. , Sivic, J., and Zisserman, A. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, pages II: 1605-1614, 2006.

[8] Tuytelaars, T., Lampert, C., Blaschko, M., and Buntine, W. Unsupervised Object Discovery: A Comparison. *IJCV*, 2010.

[9] Zhang, C., and Chen, T. An Active Learning Framework for Content Based Information Retrieval. *IEEE Transactions on Multimedia,* 4(2):260-268, 2002.