Similarity Comparisons for Interactive Fine-Grained Categorization

Catherine Wah[†] Grant Van Horn[†] Steve Branson[‡] Subhransu Maji[*] Pietro Perona[‡] Serge Belongie[♯]
[†]UC-San Diego [‡]Caltech [*]TTI at Chicago [♯]Cornell Tech
vision.ucsd.edu vision.caltech.edu ttic.edu tech.cornell.edu

Abstract

Current human-in-the-loop fine-grained visual categorization systems depend on a predefined vocabulary of attributes and parts, usually determined by experts. In this work, we move away from that expert-driven and attribute-centric paradigm and present a novel interactive classification system that incorporates computer vision and perceptual similarity metrics in a unified framework. At test time, users are asked to judge relative similarity between a query image and various sets of images; these general queries do not require expert-defined terminology and are applicable to other domains and basic-level categories, enabling a flexible, efficient, and scalable system for fine-grained categorization with humans in the loop. Our system outperforms existing state-of-the-art systems for relevance feedback-based image retrieval as well as interactive classification, resulting in a reduction of up to 43% *in the average number of questions needed to correctly classify an image.*

1. Introduction

Within the realm of visual categorization in computer vision, humans can play multiple roles. As experts, they can define a comprehensive set of semantic parts and attributes to describe and differentiate categories, as well as provide ground truth attribute values, such as for a field guide. As non-expert users of interactive classification systems [5, 38], they can also supply these attribute and part annotations. These types of systems combine machine vision algorithms with user feedback at test time in order to guide the user to the correct answer. An interactive bird species recognition system, for example, may request feedback from the user regarding a particular image, such as "Click on the beak" or "Is the wing blue?"

These attribute-based methods have several weaknesses, especially within fine-grained visual categorization. Fine-grained categories comprise the set of classes (*e.g*. Pembroke Welsh Corgi, Shiba Inu) within a basic-level category (*e.g*. dogs); each basic-level category requires its own unique, discriminative part and attribute vocabulary. Acquiring this vocabulary involves identifying an expert resource (*e.g*. a field guide) for that basic-level category. For certain categories, such as chairs or paintings, it may be difficult to produce an adequate vocabulary. Furthermore, one must obtain image- or class-level annotations for these attributes. Even if the labels were crowdsourced, each basic-level category would require a custom set of annotation tools, and building these tools is a nontrivial task.

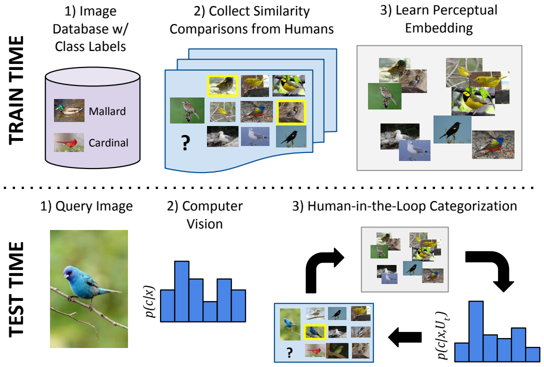
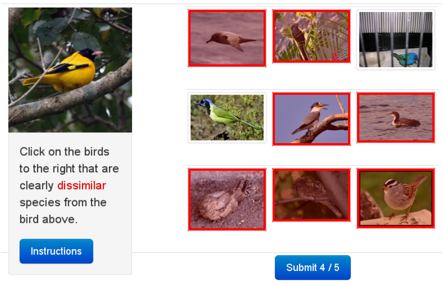

TRAIN TIME
1) Image Database w/ Class Labels
Mallard
Cardinal
2) Collect Similarity Comparisons from Humans
?
3) Learn Perceptual Embedding
TEST TIME
1) Query Image
2) Computer Vision
$p(c|x)$
3) Human-in-the-Loop Categorization
?
$p(c|x,U_t)$

Figure 1. Our interactive categorization system learns a perceptual similarity metric from human similarity comparisons on a fixed training set of images and class labels. At test time, our system leverages this learned metric, along with similarity comparisons provided by the user, to classify out-of-sample query images.

In addition, users may have difficulty understanding the domain-specific jargon used to articulate the semantic attribute vocabulary. The fixed-size vocabulary may also lack sufficient discriminative attributes for recognition. Thus, the cost in obtaining attribute vocabularies is high, making it expensive to extend an existing system to new categories.

In this work, we propose an approach to visual categorization (Fig. 1) that is based on perceptual similarity rather than an attribute vocabulary. We assume that we are provided with a fine-grained dataset of images that are annotated with only class labels. In an offline stage, we collect relative similarity comparisons between images in the dataset, and then leverage these human-provided comparisons to perform visual categorization.



Click on the birds to the right that are clearly dissimilar species from the bird above.

Figure 2. An example of the interface used for offline collection of similarity comparisons, from which we learn a similarity metric.

This similarity-based approach to interactive classification has several compelling advantages. First, we no longer require part and attribute vocabularies, which can be expensive to obtain. By eliminating the need for experts to predefine these vocabularies, we no longer constrain users by expert-defined terminology. Moreover, the continuous embedded similarity space is a richer and vastly more powerful representation than these typically fixed-size vocabularies. These factors facilitate the adaptation of an existing similarity-based system to other basic-level categories.

This similarity-based paradigm enables us to incrementally improve our computer vision models and algorithms while providing a useful service to users. Each user response collected at test time can further refine the learned similarity metrics and consequently improve performance. In addition, our flexible framework supports a variety of off-the-shelf computer vision algorithms, such as SVMs, logistic regression, and distance learning algorithms, all of which can be easily mapped into the system.

The psychology literature [35] informs us that humans judge similarity subjectively based on various universal factors that may differ from person to person; in evaluating similarity between objects in images, these factors could be based on category, pose, background, illumination, etc. Because of this, we also study how multiple general-purpose similarity metrics, with respect to universal factors such as color and shape, can be used to perform categorization.

Our visual categorization system is similar to the system in [14], with several important distinctions. While our system shares aspects of [14]'s user and display models, it uses similarity metrics that are derived from human perception of similarity rather than computer vision features, which allow us to bridge the "semantic gap" of many content-based image retrieval systems [6], including [14]. This semantic gap references the disparity between information extracted from visual data and how the user perceives and interprets that data [6]. Second, we assume that a query image is available at test time, enabling us to incorporate computer vision algorithms that are evaluated on the test image in order to initialize per-class probabilities [5]. Our system reduces human effort (as measured by the average number of questions posed to the user) by 43%, compared to an implementation of [14] that has been initialized using computer vision.

Our contributions in this work are three-fold. First, we present a efficient, flexible, and scalable system for fine-grained visual categorization that is based on perceptual similarity and combines different types of similarity metrics and computer vision methods in a unified framework. Second, we demonstrate the value in using a perceptual similarity metric over relevance feedback-based image retrieval methods and vocabulary-dependent attribute-based approaches. Lastly, we demonstrate that our system can incorporate multiple metrics, posing different forms of questions intelligently to the user at test time.

The rest of the paper is organized as follows. In Section 2, we discuss relevant work. In Section 3, we introduce our method for learning similarity metrics and describe how we integrate those metrics in our framework. We discuss implementation details in Section 4 and present our experimental results in Section 5. We conclude in Section 6.

2. Related Work

Recently, the computer vision community has seen a burst of interest in interactive classification systems [5, 38, 19], several of which build on attribute-based classification methods. Some works harvest attributes through various means [20, 12, 18, 11, 25], while others discover attributes in an automatic or interactive manner [30, 4, 8], relying on users to identify and name attributes [26, 17, 22, 23] or to provide feedback in order to improve classification [7, 28].

In contrast to these attribute-centric methods, we focus on similarity. Some recent works use similarity in feature space [18, 3]; others rely on human judgment to quantify similarity for classifying attributes or clustering categories [27, 7, 15, 21]. We instead learn a metric of perceptual similarity for categorization from relative comparisons [33, 1, 24, 34], specifically employing stochastic triplet embedding [36] in this work.

Another related area is relevance feedback-based image retrieval [32, 2, 6, 16, 40]. Some works, *e.g.*, [31], have focused on identifying nonlinear manifolds that better align with human perception; however, they do not adequately bridge the semantic gap or capture perceptual measures of similarity. In particular, our work bears similarities to the relevance feedback system presented in [14] but differs in several important ways. First, the motivating assumption in [14] is that the user possesses only a mental image or concept of a semantic category. We instead assume existence of the query image, such that we are able to incorporate computer vision at test time. Second, [14] uses a single similarity metric derived from visual features (*i.e.* GIST) rather than human perception; we conduct human experiments to generate a perceptual embedding of the data. We combine this perceptual similarity metric along with computer vision as part of a unified framework for recognition. Our system supports multiple similarity metrics and is able to trade off between these metrics at test time. To our knowledge, no other existing system combines perceptual and visual information for categorization in this integrated manner.

3. Approach

3.1. Problem Formulation

We formulate the problem as follows. Given an image x, we wish to predict the object class from C possible classes that fall within a common basic-level category, where $\mathcal{C}$ is the set of images belonging in the true object class. We do so using a combination of computer vision and a series of questions that are interactively posed to a user. Each question contains a display D of images, and the user is asked to make a subjective judgment regarding the similarity of images in D to the target image x, providing a response u.

An image x in pixel space can also be represented as a vector $\mathbf{z}$ in human-perceptual space. At train time, we are given a set of N images and their class labels $\{(x_i, c_i)\}_{i=1}^N$. We ask similarity questions to human users to learn a perceptual embedding $\{(x_i, \mathbf{z}_i, c_i)\}_{i=1}^N$ of the training data. At test time, we observe an image x and pose questions to a human user, and we obtain probabilistic estimates of $\mathbf{z}$ and c that are incrementally refined as the user answers more questions.

We also consider an extension in which similarity can be decomposed into multiple similarity metrics over A different *visual traits*. It is intended for these traits to be broadly applicable to a wide range of basic-level categories, such as similarity in terms of color, shape, or texture.

3.2. Learning Similarity Metrics

In this section, we describe how we use similarity comparisons collected from humans (Sec. 3.2.1) to learn a perceptual embedding of similarity (Sec. 3.2.2).

3.2.1 Triplet Constraints

We begin by obtaining a set of K user similarity comparisons in an offline data collection stage; more details regarding this step are discussed in Section 4.1. Each collected user response is interpreted as follows.

A user is asked to judge the similarity between a target image x and a display D that comprises a set $\mathcal{I}$ of G images. From each user response u_k, $k = 1 \dots K$, we obtain two disjoint sets: one set $\{x_{S_1}, x_{S_2}, \dots, x_{S_n}\} \in \mathcal{I}_S$ represents the images judged as similar to the query image; and $\{x_{D_1}, x_{D_2}, \dots, x_{D_m}\} \in \mathcal{I}_D$ includes all other images, such that $\mathcal{I}_D \cup \mathcal{I}_S = \mathcal{I}$. Recall that a user response for a given query image x yields two sets $\mathcal{I}_D$ and $\mathcal{I}_S$. We broadcast this to an equivalent set of (noisy) triplet constraints $\mathcal{T}^k = \{(i,j,l) | x_i \text{ is more similar to } x_j \text{ than } x_l\}$, where i is the target image, represented as x_i; j is from set $\mathcal{I}_S$; and l is drawn from set $\mathcal{I}_D$. Therefore, for each user response, we obtain nm triplet constraints in $\mathcal{T}^k$. For a display size $G = 9$, this value can range from 8 to 20 triplet constraints per user response. Constraints from each user response are then added to a comprehensive set $\mathcal{T}$ of similarity triplets.

3.2.2 Generating a Perceptual Embedding

Let $s(i,j)$ denote the perceptual similarity between two images x_i and x_j. Using $\mathcal{T}$, we wish to find an embedding $\mathbf{Z}$ of N training images $\{\mathbf{z}_1, \dots, \mathbf{z}_N\} \in \mathbb{R}^r$ for some $r \leq N$, in which triplet comparisons based on Euclidean distances are consistent with $s(\cdot,\cdot)$. In other words, we want the following to occur with high probability:

$$\|\mathbf{z}_i - \mathbf{z}_j\|_2 < \|\mathbf{z}_i - \mathbf{z}_l\|_2 \iff s(i,j) > s(i,l). \quad (1)$$

The dimensionality r is empirically chosen based on minimizing generalization error (see Sec. 5.1). We use the metric learning approach described in [36] and optimize for the embedding $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$, such that for each triplet (i,j,l) the similarity of $\mathbf{z}_i$ and $\mathbf{z}_j$ is large in comparison to the similarity of $\mathbf{z}_i$ and $\mathbf{z}_l$ according to a Student-t kernel; we refer the reader to [36] for additional details. From the learned embedding $\mathbf{Z}$, we generate a similarity matrix $S \in N \times N$ with entries:

$$S_{ij} = \exp\left(-\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{2\sigma^2}\right), \quad (2)$$

which can be directly used in our classification system. The scaling parameter σ is learned jointly with the user response model parameters (see Sec. 3.3.1). In practice, this matrix can be reduced to $S \in C \times C$, where C is the number of classes, by pooling over images in each class (see Sec. 5.1).

3.3. Human-in-the-Loop Classification

Given a test image x, the goal of our human-in-the-loop classification system is to identify the true class c as quickly as possible using a combination of computer vision and user responses to similarity questions. At each timestep t, the system intelligently chooses a display D_t of G images to show. The user provides a response u_t, selecting the image perceived to be most similar to the test image x. Let $U_t = u_1 \dots u_t$ be the set of responses obtained within timestep t. Our goal is to predict class probabilities $p(c|x, U_t)$ while exploiting the visual content of the image x and user responses U_t. We compute class probabilities by marginalizing over all possible locations $\mathbf{z}$ of image x in perceptual space:

$$p(c, U_t|x) = \int_z p(c, \mathbf{z}, U_t|x) dz \quad (3)$$

where $p(c|x, U_t) \propto p(c, U_t|x)$. Our probabilistic prediction of the location $\mathbf{z}$ and the class c becomes increasingly refined as the user answers more questions. We can further decompose $p(c, \mathbf{z}, U_t|x)$ into terms:

$$p(c, \mathbf{z}, U_t|x) \quad = \quad p(U_t|c, \mathbf{z}, x)p(c, \mathbf{z}|x) \tag{4}$$

where $p(U_t|c, \mathbf{z}, x)$ is a model of how users respond to similarity questions, and $p(c, \mathbf{z}|x)$ is a computer vision estimate.

In the following sections, we discuss the user model (Sec. 3.3.1), its efficient computation (Sec. 3.3.2), populating the display (Sec. 3.3.3), and an extension to multiple similarity metrics (Sec. 3.3.4).

3.3.1 User Response Model

We describe our probabilistic model of how users answer similarity questions as follows. We decompose user response probabilities $p(U_t|c, \mathbf{z}, x)$ as such:

$$p(U_t|c, \mathbf{z}, x) = p(U_t|\mathbf{z}) = \prod_{r=1}^{t} p(u_r|\mathbf{z}). \tag{5}$$

Here, we assume that a user's response to similarity questions depends only on the true location $\mathbf{z}$ in perceptual space and that answers to each question are independent. Recall that each similarity question comprises a display D of G images, and the user responds by selecting the index $i \in D$ of an image that is perceived to be most similar to the test image. A perfect user would deterministically choose the image x_i for which the perceptual similarity $s(\mathbf{z}, \mathbf{z}_i)$ is highest, such that:

$$p(u|\mathbf{z}) = 1[s(\mathbf{z}, \mathbf{z}_i) = \max_{j \in D} s(\mathbf{z}, \mathbf{z}_j)]. \tag{6}$$

However, real users may respond differently due to subjective differences and user error. We thus model noisy responses probabilistically, assuming that the probability that the user selects i is proportional to its similarity $s(\mathbf{z}, \mathbf{z}_i)$ to the test image x:

$$p(u|\mathbf{z}) = \frac{\phi(s(\mathbf{z}, \mathbf{z}_i))}{\sum_{j \in D} \phi(s(\mathbf{z}, \mathbf{z}_j))} \tag{7}$$

where $\phi(\cdot)$ is some customizable, monotonically increasing function. In practice, we use

$$\phi(s) = \max\left(\theta, (1 - \theta)s\right) \tag{8}$$

where θ is a learnable parameter. This model of $p(u|\mathbf{z})$ can be understood as a mixture of two distributions: with probability θ a user selects an image at random (*e.g.*, due to user error); otherwise, a user selects an image with probability proportional to its perceptual similarity. Recall from Eq 2 that $s(\mathbf{z}, \mathbf{z}_j)$ contains an additional parameter σ. Similar to [13], the parameters σ and θ are learned by maximizing the log-likelihood of a validation set of 200 non-Turker human user responses.

3.3.2 Efficient Computation

Recall that the user sequentially answers a series of similarity questions $U_t = u_1 \ldots u_t$. In this section, we derive an efficient algorithm for updating class probability estimates $p(c|x, U_t)$ in each timestep t.

Let w_k^t be shorthand for the probability $p(c_k, \mathbf{z}_k, U_t|x)$:

$$w_k^t \quad = \quad \left(\prod_{r=1}^{t} p(u_r|\mathbf{z}_k) \right) p(c_k, \mathbf{z}_k|x) \tag{9}$$

where k enumerates images in the training set. Each weight w_k captures how likely location $\mathbf{z}_k$ is the true location $\mathbf{z}$. Note that w_k^{t+1} can be efficiently computed from w_k^t as:

$$w_k^{t+1} \quad = \quad p(u_{t+1}|\mathbf{z}_k)w_k^t = \frac{\phi(S_{ik})}{\sum_{j \in D} \phi(S_{jk})} w_k^t \tag{10}$$

where i is the selected image at $t + 1$, S_{ij} is an entry of the similarity matrix (Sec. 3.2.2), and $w_k^0 = p(c_k, \mathbf{z}_k|x)$. To estimate class probabilities, we approximate the integral in Eq 3 as the sum over training examples:

$$p(c, U_t|x) \quad \approx \quad \frac{1}{N} \sum_{\substack{k=1\ldots n, \\ c_k = c}} p(c_k, \mathbf{z}_k, U_t|x). \tag{11}$$

By the definition of w_k^t and normalizing probabilities, it follows that $p(c|x, U_t)$ is the sum of the weights of training examples of class c:

$$p(c|x, U_t) = \frac{\sum_{k, c_k = c} w_k^t}{\sum_k w_k^t}, \tag{12}$$

resulting in an efficient algorithm where we maintain weights w_k^t for each training example: (1) we initialize weights $w_k^0 = p(c_k, \mathbf{z}_k|x)$ (estimated using computer vision; see Sec. 3.4); (2) we update weights when the user answers a similarity question (Eq 10); and (3) we update per-class probabilities (Eq 12).

3.3.3 Choosing Which Images to Display

Recall that at each timestep, our system intelligently poses a similarity question by selecting a display D of G images. We wish to choose the set of images that maximizes expected information gain. We follow the procedure used by Ferecatu and Geman [14], which defines an efficient approximate solution for populating this display. We group the images into equal-weight clusters, where each image possesses mass w_k^t. This ensures that each image in the display is equally likely to be clicked, maximizing the information gain in terms of the entropy of $p(c, \mathbf{z}_k, U_t|x)$. Given the clustering of images, we pick the image within the cluster with the highest mass for the display using an approximate solution. We refer the reader to [10, 14] for additional details. A similar procedure can be used to instead pick a set of G classes to display, assigning each class a mass $\sum_{k,c_k=c} w_k^t$, maximizing the information gain in terms of the entropy of $p(c|x, U_t)$.

3.3.4 Extension to Multiple Similarity Metrics

Our system can support the use of multiple similarity metrics S^a, $a \in 1 \ldots A$ that are represented at test time as different questions, where we direct the user's attention to specific visual traits. At train time, we obtain a separate embedding $\mathbf{Z}^1 \ldots \mathbf{Z}^A$ for each trait (using similarity questions that are targeted toward a specific trait), yielding multiple similarity matrices $S^1 \ldots S^A$.

At test time at each timestep t, we pick both a trait a and display of images D that is likely to provide the most information gain. This amounts to finding the trait that can produce the most balanced clustering according to the current weights w_k^t. Computation of updated class probabilities occurs identically to the procedure described in Section 3.3.2, with a slightly modified update rule that replaces Eq 10:

$$w_k^{t+1} \quad = \quad p(u_{t+1}|\mathbf{z}_k^a) w_k^t = \frac{\phi(S_{ik}^a)}{\sum_{j \in D} \phi(S_{jk}^a)} w_k^t. \quad (13)$$

Here, we update weights w_k^{t+1} according to the similarity matrix S^a of the selected trait a.

3.4. Incorporating Computer Vision

Recall from Eq 4 that we would like to train an estimator for $p(c, \mathbf{z}|x)$, the probability that an observed image x belongs to a particular class c and location $\mathbf{z}$ in perceptual space. In practice, our human-in-the-loop classification algorithm (as described in Sec. 3.3.2) only requires us to estimate $w_k^0 = p(c_k, \mathbf{z}_k|x)$ for training examples $k = 1...N$ rather than for all possible values of $\mathbf{z}$. In this section, we show how off-the-shelf computer vision algorithms such as SVMs, boosting, logistic regression, and distance learning algorithms can be mapped into this framework. We also discuss novel extensions for designing new algorithms that are more customized to the form of $p(c, \mathbf{z}|x)$. For each such method, we describe the resulting computation of w_k^0.

No Computer Vision: If no computer vision algorithm is available, then we have no information toward predicting c or $\mathbf{z}$ based on observed image pixels x. As such, we assume each location $\mathbf{z}_k$ is equally likely:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) = \frac{1}{N}. \quad (14)$$

Classification Algorithms: Classification algorithms such as SVMs, boosting, and logistic regression produce a classification score that can be adapted to produce a probabilistic output $p(c|x)$. They are otherwise agnostic to the prediction of $\mathbf{z}$. We thus assume that $\mathbf{z}_i$ and $\mathbf{z}_j$ are equally likely for examples of the same class $c_i = c_j$:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) = \frac{1}{N_{c_k}} p(c_k|x) \quad (15)$$

where N_c is the number of training images of class c. We learn parameters for $p(c|x)$ on a validation set [29].

Distance-Based Algorithms: Non-parametric methods (e.g., nearest neighbor and distance-learning methods) can be adapted to produce a similarity $s(x_k, x)$ between x and the k_{th} training example (computed using low-level image features) but are otherwise agnostic to class:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) \propto s(x_k, x). \quad (16)$$

A Gaussian kernel $s(x_k, x) = \exp\{-d(x_k, x)/\sigma\}$ is commonly used, where $d(x_k, x)$ is a distance function and σ is estimated on a validation set. Note that due to normalization in Eq 12, using an unnormalized probability does not affect correctness.

Pose-Based Classification Algorithms: Note that the above classification and distance-based algorithms are sub-optimal due to not exploiting information in $\mathbf{z}_k$ and c, respectively. We consider a simple extension to help remedy this. We obtain a perceptual pose embedding $\mathbf{Z}^o$ of the training data using pose similarity questions (see Sec. 3.2.2), then cluster training examples $\mathbf{z}_1^o ... \mathbf{z}_N^o$ using k-means into K discrete poses. Let o_i be the pose index of the i_{th} example. We train a separate multiclass classifier for each pose o, obtaining a pose-conditioned class estimator for $p(c|x, o)$. We similarly train a multiclass pose classifier that estimates pose probabilities $p(o|x)$. We assume our classifiers give us information about $\mathbf{z}$ through pose labels o but are otherwise agnostic to the prediction of $\mathbf{z}$:

$$w_k^0 = p(c_k, \mathbf{z}_k|x) = \frac{1}{N_{c_k, o_k}} p(c_k|x, o_k) p(o_k|x) \quad (17)$$

where N_{co} is the number of training examples of class c and pose o. At test time, we have the option of asking a mixture of class and pose similarity questions.

4. Implementation

4.1. Dataset and Data Collection

We perform experiments on *CUB-200-2011* [39], which contains 200 bird classes with roughly 60 images per class. We maintain the training/testing split—only training images are seen in the data collection phase and are used to generate the embedding. Test images are considered as out-of-sample input to the interactive categorization system.

To collect the similarity comparisons, we created an interface (Fig. 2) that displays a reference image along with a grid of 3 × 3 images. Amazon Mechanical Turk workers are asked to select all the images in the grid that clearly belong to a different species, as compared to the reference image. Images for each task are sampled at the category level without replacement, such that no two images belong to the same category. Additional observations regarding how the collected data impacts the embedding generation are discussed in Section 5.1.

4.2. Features and Learning

We use multiclass classifiers to initialize $p(c, \mathbf{z}|x)$, extracting color/grayscale SIFT features and color histograms with VLFEAT [37] that were combined with spatial pyramids. We trained 1-vs-all SVMs using LIBLINEAR [9], achieving an average classification accuracy of 19.4% on the test set. The classification scores are used to update w_k^0 according to Eq 15. At test time, we display a ranked list of classes based on the posterior probabilities, from which users can verify the class of the input image.

5. Experiments

5.1. Embedding Generation

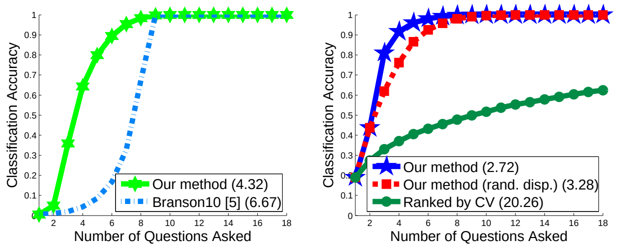
Using a set of triplets generated from our collected similarity comparisons, we are able to learn an embedding (Fig. 4(a)) of N nodes, where N=200 is the number of classes. To better understand the tradeoff between dimensionality r and embedding accuracy, we compute the generalization error as we sweep over the number of dimensions. The generalization error measures the percentage of held-out similarity triplets satisfied in three-fold cross validation. With this method, we empirically estimate r=10 as sufficient for minimizing generalization error.

In Figure 4(a), various clusters of classes are highlighted. We observe that visually similar classes tend to belong to coherent clusters within the embedding, for example, the gulls, large black birds, and small brown striped birds. However, we also note that certain species that are dissimilar to the other birds tend to fall in their own cluster, towards the upper left portion of the embedding.

An embedding at the category level does not characterize intraclass variation, which can be high due to differences in gender, age, season, etc. Instead, this is handled through the noisy user model (Eq 7). While our method does not inherently require it, learning a similarity metric at the category level requires much fewer annotations and still gives a reasonable metric of similarity. In our experiments, we used roughly $93,000$ triplets out of a possible 8 million to generate a category-level embedding. At the instance level, this would be equivalent to collecting over 2 billion triplets.

5.2. Interactive Categorization

We present our results for interactive classification using the learned perceptual metric for class similarity in Figures 3 and 4(b). Qualitative examples of results are presented in Figure 5(a). At test time, a user is shown a display of 3 × 3 images and asked to select the bird that is most similar to the input class. The input image is drawn from the test set, and the display images are drawn strictly from the pool of training images. As such, the system does not possess prior knowledge of perceptual similarity between a given input image and any possible display of images. We use simulated user responses, which facilitates comparison to previous work as well as allows us greater flexibility in running experiments. Playback simulations based on real human responses are common in human-in-the-loop work [5, 38, 26, 27, 28] as they allow algorithmic and parameter setting choices to be explored without rerunning human experiments.

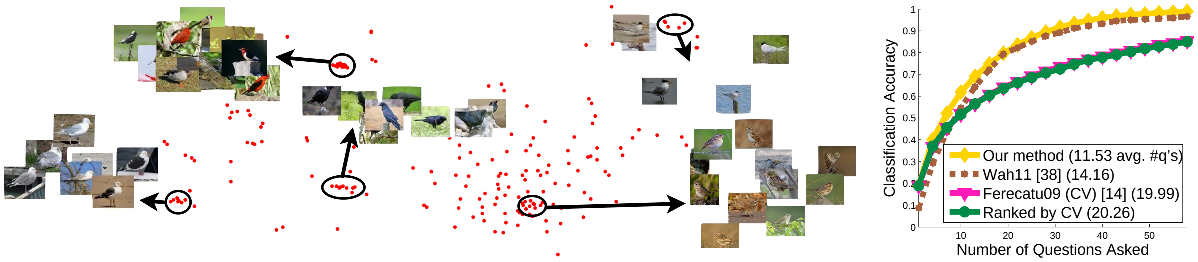


(a) No computer vision (b) With computer vision

Figure 3. *Deterministic users*. We report the average number of questions asked per test image in parentheses for each method. 3(a): Our similarity-based approach requires fewer questions (4.32 vs. 6.67) than [5], which uses attributes. 3(b): Our display mechanism reduces user effort, as compared to randomly generated grids of images and a baseline based on the ranked classification scores.

In our experiments, we measure classification accuracy as a function of the number of questions or displays the user has seen. We use the same experimental setup and evaluation criteria as [38], assuming that humans can verify the highest probability class perfectly and can stop the system early. Performance is measured as the average number of questions that a user must answer per test image to classify it correctly. Different types of questions (similarity, attribute, or part-based) may incur varying amounts of cognitive effort on the user's part, which may be reflected in differing amounts of time to answer a single question. As our test-time user responses are simulated, we compare performance based on the number of questions posed.

Similarity comparisons are advantageous compared to attribute questions. In Figures 3(a) and 3(b), we show the effects of not using and using computer vision, respectively. We observe performance using deterministic (perfect) users (Eq 6) who are assumed to respond in accordance with the learned similarity metric. For a direct comparison to attribute-based approaches, we compare our method to the setting in which users answer attribute questions deterministically in accordance with expert-defined class-attribute



(a) Category-level similarity embedding

(b) Noisy users

Figure 4. 4(a): A visualization of the first two dimensions of the 200-node category-level similarity embedding. Visually similar classes tend to belong to coherent clusters (circled and shown with selected representative images). 4(b): *Simulated noisy users*. Our method outperforms a part and attribute-based interactive classification system [38] as well as the relevance feedback-based image retrieval system described in [14], which has been modified to utilize computer vision in initializing per-class probabilities for fairness of comparison.

values, as reported in [5]. We are able to reduce the average number of questions needed by 2.4.

Computer vision reduces the burden on the user. We note a similar trend when computer vision is incorporated at test time (Fig. 3(b)), in which users take an average of 2.7 questions per image. The addition of computer vision (Sec. 3.4) reduces the number of questions a user must answer in order to classify an image by 1.6 (Fig. 3(a)).

Intelligently selecting image displays reduces effort. We compare performance for two versions of our method: the first intelligently populates each display (Sec. 3.3.3) and the second randomly generates a display of images at each question. Using our display model, we observe that 2.7 questions are required on average, compared to 3.3 questions using a random display. We also compare to a baseline derived from classification scores [Ranked by CV], in which the user moves down the ranked list of classes one at a time to verify the correct class. With our model, we reduce the average number of questions from 20.3 to 2.7.

Our system is robust to user noise. In reality, assuming deterministic users is impractical, as users are likely to have subjective differences in their perceptions of similarity. To account for this, we incorporate a user response model that accounts for real human behavior (see Sec. 3.3.1). Using a validation set of query images, we pose similarity questions to real human users and estimate the parameters of a noisy user response $p(u|\mathbf{z})$ with the collected responses.

In our experiments, we simulate noisy user behavior at test time by randomly selecting answers according to the distribution $p(u|\mathbf{z})$. We compare performance directly to the results presented in [38], a system that uses part-localized computer vision algorithms as well as user feedback via attribute and part-click questions, obtaining a reduction of 2.6 questions on average (Fig. 4(b)).

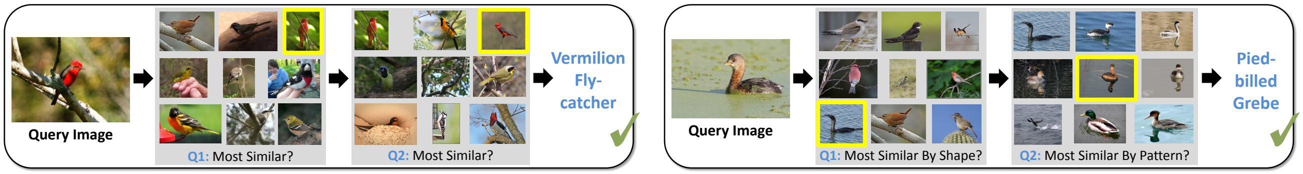
We also improve performance significantly over an implementation of [14] that uses a similarity metric generated from the L1 distances of concatenated feature vectors (see Sec. 4.2). For a fair comparison, the system in [14] is modified to use computer vision in initializing the per-class probabilities, as the query image is provided. We note that the use of the L1 distance-based metric is unable to adequately capture perceptual similarity, resulting in a high average number of questions needed for categorization.

Method	Avg # Questions
CV, Color Similarity	2.70
CV, Shape Similarity	2.67
CV, Texture Similarity	2.67
CV, Color/Shape/Texture Similarity	**2.64**
No CV, Color/Shape/Texture Similarity	4.21

Table 1. Results using multiple synthetic similarity metrics with deterministic users. See Section 5.2.1 for additional details.

5.2.1 Using Multiple Similarity Metrics

We demonstrate a proof-of-concept that our human-in-the-loop system can utilize multiple similarity metrics. Ideally, these metrics would be generated from human responses on visual trait similarity; however, due to the time expense of collecting new similarity datasets, we simulate perceptual spaces using *CUB-200-2011* attribute annotations. The attribute vectors used to synthesize these metrics are averaged over multiple human responses to attribute questions, and therefore capture some perceptual measurements. Similarity metrics are generated for certain universal traits by comparing category-level binary attribute vectors; for example, the color trait is represented as a vector of the color-related attributes. The pairwise Euclidean distances between binary attribute vectors are used to generate a similarity matrix. The observed traits—color, shape, and texture—are universal enough to be useful in describing a range of basic-level categories. In this way, these traits would not necessitate the creation of an attribute vocabulary for a new basic-level category. We present our results using deterministic users in Table 1 and Figure 5(b).



(a) Category similarity metric

(b) Multiple synthetic similarity metrics

Figure 5. Qualitative results using metrics 5(a) learned from AMT workers and 5(b) synthesized from binary attribute vectors.

6. Conclusion

We have presented an efficient approach to interactive fine-grained categorization that does not rely on experts for attribute vocabularies and is cost-effective to deploy for new basic-level categories. As users answer similarity questions for new query images, we can augment the training set and regenerate the perceptual similarity metric, enabling the system to iteratively improve as more responses are collected. Future work could involve using these perceptual embeddings to induce attributes, parts, taxonomies, etc., which may be of educational value to a user. In addition, as often there exists no ground truth relative similarity judgment, it would be of interest to the computer vision community to determine best practices of eliciting consistent user similarity comparisons.

7. Acknowledgments

The authors thank the reviewers for their helpful feedback. This work is supported by the Google Focused Research Award and the National Science Foundation Graduate Research Fellowship under Grant No. DGE0707423.

References

[1] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. J. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR*, 2005. 2

[2] C. C. Aggarwal. Towards meaningful high-dimensional nearest neighbor search by HCI. In *ICDE*, 2002. 2

[3] B. Babenko, S. Branson, and S. Belongie. Similarity metrics for categorization. In *CVPR*, 2009. 2

[4] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010. 2

[5] S. Branson et al. Visual recognition with humans in the loop. In *ECCV*, 2010. 1, 2, 6, 7

[6] R. Datta et al. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 2008. 2

[7] J. Donahue and K. Grauman. Annotator rationales for visual recognition. In *ICCV*, 2011. 2

[8] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering Localized Attributes for FGVC. In *CVPR*, 2012. 2

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR. *JMLR*, 2008. 6

[10] Y. Fang and D. Geman. Experiments in mental face retrieval. In *AVBPA*, 2005. 4

[11] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2

[12] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 2

[13] M. Ferecatu and D. Geman. Interactive search for image categories by mental matching. In *ICCV*, 2007. 4

[14] M. Ferecatu and D. Geman. A statistical framework for category search from a mental picture. *TPAMI*, 2009. 2, 4, 7

[15] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, 2011. 2

[16] J. Hare et al. Bridging the semantic gap in multimedia information retrieval. In *ESWC*, 2006. 2

[17] A. Kovashka, S. Vijayanarasimhan, and K. Grauman. Actively selecting annotations among objects and attributes. In *ICCV*, 2011. 2

[18] N. Kumar et al. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 2

[19] N. Kumar et al. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012. 2

[20] C. Lampert et al. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2

[21] E. Law et al. Human computation for attributes and attribute values acquisition. In *FGVC Workshop*, 2011. 2

[22] S. Maji. Discovering a lexicon of parts and attributes. In *ECCV Workshop on Parts and Attributes*, 2012. 2

[23] S. Maji and G. Shakhnarovich. Part annotations via pairwise correspondence. In *Human Computation Workshop*, 2012. 2

[24] B. McFee and G. Lanckriet. Metric learning to rank. In *ICML*, June 2010. 2

[25] M. Palatucci et al. Zero-Shot Learning with Semantic Output Codes. In *NIPS*, 2009. 2

[26] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011. 2, 6

[27] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2, 6

[28] A. Parkash and D. Parikh. Attributes for classifier feedback. In *ECCV*, 2012. 2, 6

[29] J. Platt. Probabilities for SV machines. In *NIPS*, 1999. 5

[30] M. Rohrbach et al. What helps where – and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010. 2

[31] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. In *IJCV*, volume 40, 2000. 2

[32] Y. Rui et al. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE TCSV*, 8(5), 1998. 2

[33] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2003. 2

[34] O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. Kalai. Adaptively learning the crowd kernel. In *ICML*, 2011. 2

[35] A. Tversky. Features of similarity. *Psychological rev*, 1977. 2

[36] L. van der Maaten and K. Weinberger. Stochastic triplet embedding. In *MLSP*, 2012. 2, 3

[37] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library. http://www.vlfeat.org/, 2008. 6

[38] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass rec. and localization with humans in the loop. In *ICCV*, 2011. 1, 2, 6, 7

[39] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-UCSD Birds-200-2011. Technical report, Caltech, 2011. 5

[40] X. Zhou and T. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 2003. 2