Int J Comput Vis (2014) 108:3–29
DOI 10.1007/s11263-014-0698-4

The Ignorant Led by the Blind: A Hybrid Human–Machine Vision System for Fine-Grained Categorization

Steve Branson · Grant Van Horn · Catherine Wah · Pietro Perona · Serge Belongie

Received: 7 March 2013 / Accepted: 8 January 2014 / Published online: 20 February 2014
© Springer Science+Business Media New York 2014

Abstract We present a visual recognition system for fine-grained visual categorization. The system is composed of a human and a machine working together and combines the complementary strengths of computer vision algorithms and (non-expert) human users. The human users provide two heterogeneous forms of information object part clicks and answers to multiple choice questions. The machine intelligently selects the most informative question to pose to the user in order to identify the object class as quickly as possible. By leveraging computer vision and analyzing the user responses, the overall amount of human effort required, measured in seconds, is minimized. Our formalism shows how to incorporate many different types of computer vision algorithms into a human-in-the-loop framework, including standard multiclass methods, part-based methods, and localized multiclass and attribute methods. We explore our ideas by building a field guide for bird identification. The experimental results demonstrate the strength of combining ignorant humans with poor-sighted machines the hybrid system achieves quick and accurate bird identification on a dataset containing 200 bird species.

Communicated by M. Hebert.

Electronic supplementary material The online version of this article (doi:10.1007/s11263-014-0698-4) contains supplementary material, which is available to authorized users.

S. Branson (✉) · P. Perona
Caltech, Pasadena, CA , USA
e-mail: sbranson@caltech.edu

G. Van Horn · C. Wah · S. Belongie
University of California, San Diego,
9500 Gilman Dr, La Jolla, CA 92093, USA

Keywords Fine-grained categorization · Human-in-the-loop · Interactive · Parts · Attributes · Crowdsourcing · Deformable part models · Pose mixture models · Object recognition · Information gain · Birds

1 Introduction

Fine-grained categorization, also known as subordinate categorization in psychology literature (Rosch 1999; Mervis and Crisafi 1982; Biederman et al. 1999), has emerged in recent years as a problem of great interest to the computer vision community, with applications including species identification for animals (Wah et al. 2011; Liu et al. 2012; Khosla et al. 2011), plants (Kumar et al. 2012), flowers (Nilsback and Zisserman 2008) and insects (Larios et al. 2010) as well as classification of man-made objects such as vehicle makes and models (Stark, et al. 2012) and architectural styles (Maji and Shakhnarovich 2012). Fine-grained visual categories lie in the space between basic (or entry) level categories (Rosch et al. 1976) (e.g., the 20 classes from PASCAL VOC including motorbikes, dining tables, etc.) and identification of individuals (e.g., face or fingerprint biometrics). As the visual distinctions among fine-grained categories are often quite subtle, a given general-purpose tool popular for basic-level category recognition can be rendered a rather blunt instrument in the fine-grained case.

While a layperson can recognize entry level categories like bicycles or birds immediately, fine-grained categories are difficult for untrained humans. They are typically recognized only by experts. This work arises from a key realization: while fine-grained visual categorization is difficult for both humans and machines, humans and machines have radically different strengths and weaknesses. Humans are able to detect and broadly categorize objects, even when they do

not recognize them. They can localize basic shapes and parts, and recognize colors and materials (see Figs. 1, 2). Human errors arise primarily because people have (1) limited experiences and memory and (2) subjective and perceptual differences. In contrast, computers can run deterministic software and aggregate large databases of information. They excel at memory intensive problems like recognizing movie posters or cereal boxes but struggle with objects that are textureless, immersed in clutter, highly articulated or non-trivially deformed. This suggests that a visual system composed of a human and a machine can carry out the task, and do so efficiently, by combining the strengths of each; this requires a dynamic collaboration between the two agents.

With the goal of developing a combined human and machine system for visual classification, we introduce models and algorithms that account for errors and inaccuracies of computer vision algorithms (part localization, attribute detection and object classification) and ambiguities in multiple forms of human feedback (perception of part locations, attribute values and class labels). An example human-in-the-loop system is depicted in Fig. 3, where a picture of an unknown bird species is identified using a combination of computer vision and intelligently selected questions (*e.g.*, *click on the beak*, *what is the primary color of the bird?*, *etc.*). Our approach combines the complementary strengths of humans and computers for these different modalities by optimizing a single principled objective function: minimizing the expected amount of time to complete a given classification task.

Our models and algorithms combine all such sources of information, including human responses to part-click, binary, multiple choice and multi-select questions, into a single principled framework. We have implemented a practical real-time system for bird species identification on a 200-category dataset. Recognition and pose registration can be achieved automatically using computer vision; the system can also incorporate human feedback when computer vision is unsuc-

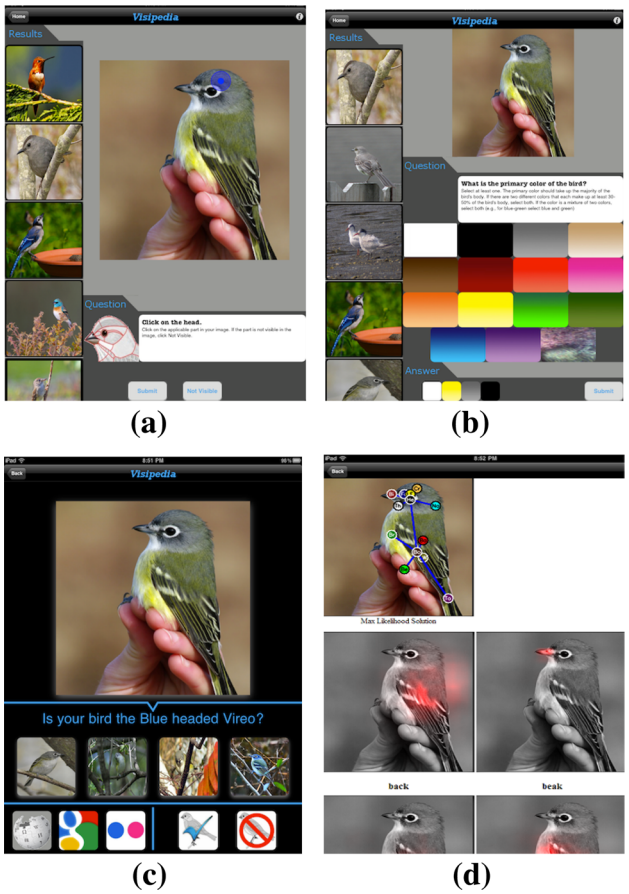


Fig. 1 Screen capture of an ipad app for bird species recognition. A user takes a picture of a bird she wants to recognize, which is uploaded to a server. The server runs computer vision algorithms to localize parts and predict bird species. The computer system intelligently selects a series of questions to ask that are designed to reduce its uncertainty about the predicted bird species as quickly as possible. (**a**) The system poses the question *click on the head*? The user's click response is used to refine part location and class probability estimates. (**b**) The system chooses another *what is the primary color of the bird?* (**c**) The system thinks that the bird is a *Blue-headed Vireo*. (**d**) Debugging output of the algorithms shows detected part locations and part probability maps (Color figure online)

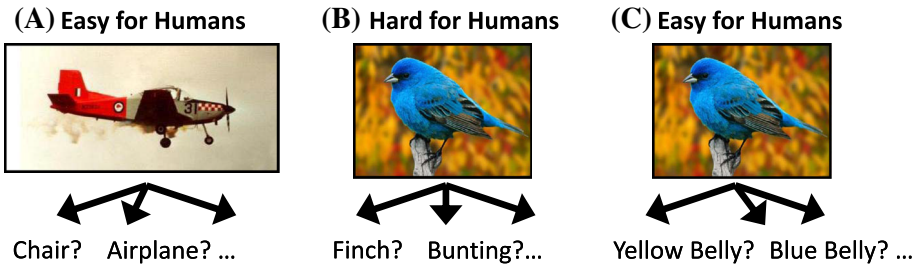


Fig. 2 Examples of classification problems that are easy or hard for humans. While basic-level category recognition (*left*) and recognition of low-level visual attributes (*right*) are easy for humans, most people struggle with fine-grained categories (*middle*). By defining categories in terms of low-level visual properties, hard classification problems can be turned into a sequence of easy ones

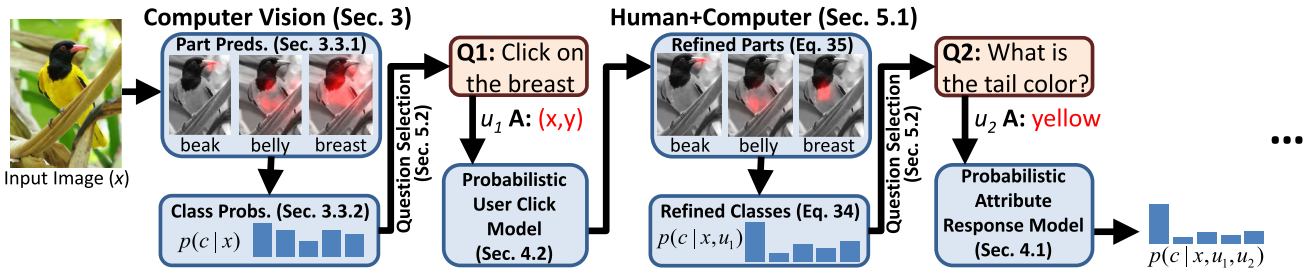


Fig. 3 System overview. Our system uses a combination of computer vision and interactive feedback to recognize bird species. It begins by running computer vision algorithms to localize parts and predict bird species. The system intelligently selects a question *click on the breast* that it believes will maximally reduce its ambiguity about its species prediction. The user's response significantly improves the localization of both the breast and the belly, which refines class estimates. The system chooses another question, *what is the tail color*? The user's answer of *yellow* is used to further refine class probability estimates. The process continues until the user stops the interface (Color figure online)

cessful by intelligently posing questions to human users (see Fig. 3).

1.1 Contributions

This paper makes four contributions:

1. A hybrid human–machine vision system for subordinate categorization. The design includes a GUI, a method for measuring the statistics of human-provided attributes for each category, a method for estimating class probabilities from computer vision measurements and human-provided information, and an algorithm for selecting the most informative questions that should be posed to the human users.
2. A computer vision system for automated fine-grained categorization. Our algorithms can localize and classify objects on a 200-class dataset in a fraction of a second, using detectors that are shared among classes. Our fully automated computer vision algorithms significantly outperform earlier methods on CUB-200-2011 (Wah et al. 2011)
3. A formal model for evaluating the usefulness of different types of human input. We introduce fast algorithms that are able to predict the informativeness of 312 binary questions, 29 multiple choice and multi-select questions, and 15 part click questions in a fraction of a second.
4. A thorough experimental comparison of a number of methods for optimizing human input. We include results of a real world study of 27 human subjects using our bird identification tool.

We have implemented our algorithms into practical tools for bird species identification, including a web-based identification tool and an iPad app (see Fig. 1). The design of our system is modular and can be used in conjunction with a wide variety of computer vision algorithms. A visualization of the different components of our system is shown in Fig. 3.

1.2 Differences from Earlier Work

This article consolidates earlier work published in ECCV 2010 (Branson et al. 2010) and ICCV 2011 (Wah et al. 2011), but also contains a significant amount of new results and material:

- We performed more extensive experiments, including a user study of people using a realtime web-based version of our system to identify birds.
- Performance of computer vision algorithms has been significantly improved, both in terms of part localization and multiclass species classification [improving classification accuracy on CUB-200-2011 (Wah et al. 2011) from 10.3 % (Wah et al. 2011) to 56.8 %]
- We added support for multiple choice and multi-select questions, leading to significant reductions in human time over binary questions [along with new computer vision algorithms, average time to classify species CUB-200-2011 has been reduced from 58.4 s (Wah et al. 2011) to 20.53 s]
- Additional implementation details have been added throughout the paper, including details on how we convert computer vision systems to probabilities, and formalized details of how to put a wider array of computer vision algorithms into a human-in-the-loop framework
- We added supplementary material with additional details on improved computer vision algorithms (including new pose clustering techniques, more sophisticated features, and structured learning algorithms), dataset statistics, qualitative examples, videos of our user interface, and analysis of which questions were selected by our system.

1.3 Paper Structure

The structure of the paper is as follows. In Sect. 2, we review related work. We define the problem and describe different types of computer vision algorithms for multiclass recognition, part localization, and attribute-based classification in

Sect. 3. In Sect. 4, we introduce our models for human annotators based on crowdsourced data collection. We then describe our approach to combine human and machine computation for the problem of localized recognition in Sect. 5. We present our experimental results and the findings of our human user study in Sect. 6. Finally we conclude and discuss future work in Sect. 7.

2 Related Work

2.1 Fine-Grained Categorization

Fine-grained visual categorization (FGVC) is a challenging problem that has recently become a popular topic in computer vision. Applications include recognizing different species of leaves (Kumar et al. 2012; Belhumeur et al. 2008), flowers (Nilsback and Zisserman 2006, 2008), dogs (Parkhi et al. 2011; Liu et al. 2012; Parkhi et al. 2012; Khosla et al. 2011), birds (Branson et al. 2010; Farrell et al. 2011; Wah et al. 2011; Zhang et al. 2012; Lazebnik et al. 2005), and stonefly larvae (Martınez-Munoz et al. 2009; Larios et al. 2010). Each of these can be seen as interesting scientific applications with a significant appeal to a specific demographic of users, enthusiasts, or citizen scientists. In conjunction with this, many new FGVC datasets have emerged with richer annotations, such as CUB-200-2011 (Wah et al. 2011) (birds with parts and attributes), Columbia Dogs With Parts (Liu et al. 2012), Leeds Butterflies (Wang et al. 2009) (segmentations and text descriptions), Oxford-IIIT Pets (Parkhi et al. 2011, 2012) (cats and dogs with segmentations and bounding boxes), and Stanford Dogs (Khosla et al. 2011) (bounding boxes).

Most research in FGVC is related to finding less lossy features, models, or representations to deal with tightly related categories. The work of (Yao et al. 2011, 2012) and (Martınez-Munoz et al. 2009) relates to learning features that go beyond traditional codebook-based methods in object recognition. (Nilsback and Zisserman 2008) and (Chai et al. 2011, 2012) introduce techniques that improve ROI for feature extraction by simultaneously segmenting and recognizing FGVCs. Other methods focus on incorporating part/pose detectors that supplant or augment bag-of-words methods by allowing for more strongly localized visual features Farrell et al. (2011); Wah et al. (2011); Parkhi et al. (2011); Zhang et al. (2012); Liu et al. (2012); Parkhi et al. (2012). Most of these methods exploit new types of annotation. The work of (Farrell et al. 2011; Zhang et al. 2012) explores different methods for pose normalization using Poselets, including an original method that is based on 3D volumetric primitives.

The computer vision component of our algorithms is related to this area; we employ part/pose detection that is based on mixtures of deformable part models (Yang and Ramanan 2011; Wah et al. 2011; Branson et al. 2011), a model that is similar in its representational power to Poselets. We chose this method because it is popular and high-performing, while also being easily formalizable and understandable as a probabilistic model. This allowed us to mix our detection models with new types of human feedback and localized attribute detection techniques. Despite this, we believe that similar types of interactive methods could be incorporated with other pose normalization schemes.

2.2 Human-in-the-Loop Methods

FGVC is difficult for both humans and computers. An interactive algorithm that assists a human in discovering the true class is useful and preferable to a fully automatic yet error-prone algorithm. Human-in-the-loop methods have recently experienced a strong resurgence in popularity. (Parikh and Zitnick 2011a,b) introduced an innovative human debugging framework, using human experiments to help diagnose bottlenecks in computer vision research. This work is similar in spirit to our work in that it involves comparing the visual capabilities of humans and computers.

A number of exciting active learning algorithms that incorporate new types of human interactivity have come about in recent years (Vijayanarasimhan and Grauman 2009, 2011; Donahue and Grauman 2011; Vondrick and Ramanan 2011; Settles 2008; Parkash and Parikh 2012; Branson et al. 2011). In the domain of active learning, our work is most similar to the work of (Vijayanarasimhan and Grauman 2009) on cost-sensitive active learning. Our approach is similar in that we also optimize an information-theoretic criterion to actively choose a certain type of annotation based on its expected annotation time. The main difference is that our work pertains to active testing (*i.e.*, incorporating similar types of interactive feedback at classification time instead of during learning), and we develop interactive querying strategies for types of annotations not considered in earlier work (*i.e.*, attribute and part localization annotations).

Interactive methods for generating vocabularies of parts or attributes (Maji 2012; Parikh andGrauman 2011; Duan et al. 2012) and incorporating annotator rationales (Donahue and Grauman 2011), and runtime interactive computer vision systems for segmentation and tracking (Wu and Yang 2006; Rother et al. 2004; Levin et al. 2007; Vondrick et al. 2010) are all interesting related lines of research that apply to applications that are not considered in this work (*i.e.*, we specifically address the area of hybrid human–computer classification).

Our method bears the most resemblance to relevance feedback methods in content-based image retrieval (CBIR) (Rasiwasia et al. 2007; Ferecatu and Geman 2007; Zhou and Huang 2003; Lu et al. 2000; Cox et al. 2000; Parikh and

Grauman 2013), where human feedback is used to interactively refine the result of image search. Our method shares the same basic objective: combining computer vision with human feedback to solve some task as quickly as possible. As such, components of our method build off techniques that were developed earlier in relevance feedback literature—in particular, the use of attributes (or some semantic categorical space) as a vehicle for communicating with humans (Rasiwasia et al. 2007; Lu et al. 2000; Kumar et al. 2008; Douze et al. 2011; Parikh and Grauman 2013) and the use of information theoretic techniques to select which type of query to pose to the human user (Ferecatu and Geman 2007, 2009; Cox et al. 2000) (see Sect. 2.3 for further discussion). The main distinguishing feature of our approach is the development of a more extensive hybrid human–computer model for different types of computer vision algorithms for object recognition, object detection, part localization, and attribute prediction (*i.e.*, beyond similarity functions and classifiers based on low-level features), and how these different types of algorithms naturally interact with different types of user input.

2.3 Active Testing

Our methodology for selecting which questions to pose to human users is an instance of active testing (Geman and Jedynak 1993, 1996; Tsiligkaridis et al. 2013; Jedynak et al. 2012), where a sequence of questions are chosen at runtime to minimize as much uncertainty as possible about some prediction task (*e.g.*, consider the *Twenty Questions Game*). Similar to decision trees (Quinlan 1993), the criterion for choosing the next question is information theoretic; however, unlike decision trees, questions are chosen on-the-fly at runtime—precomputed decision trees would be intractably large (*i.e.*, due to an excessively large branching factor or depth as a result of more complex sources of information).

Active testing has been applied to computer vision to speedup object localization and tracking problems (Geman and Jedynak 1993, 1996; Sznitman and Jedynak 2010; Sznitman et al. 2011), where the active testing system sequentially chooses locations to evaluate a detector (rather than brute force evaluate a sliding window detector), iteractively refining its belief of where the object is located. The main difference between these methods and ours is the use of a hybrid model where computer vision estimates are augmented with questions that are posed interactively to humans (as opposed to a computer). Ferecatu et al. (Ferecatu and Geman 2007, 2009; Fang and Geman 2005) applied active testing to image retrieval with relevance feedback, developing a a system that intelligently selects similarity questions to pose to human users. The main difference between this approach and ours is the incorporation of computer vision at runtime [*i.e.*, (Ferecatu and Geman 2007) considers the "mental matching" problem where no image is present at runtime].

2.4 Parts and Attributes

Methods based on parts (Felzenszwalb et al. 2008; Felzenszwalb and Huttenlocher 2002; Bourdev and Malik 2009; Ott and Everingham 2011; Yang and Ramanan 2011) and attributes (Farhadi et al. 2009; Lampert et al. 2009; Kumar et al. 2009; Farhadi et al. 2010; Wang and Forsyth 2009; Parikh and Grauman 2011) have both become popular, mainstream topics in computer vision research. An interesting component of FGVC problems is that similarities between classes are exploitable for transfer learning or model sharing methods (*i.e.*, different bird species share the same types of parts and attributes). FGVC methods that incorporate a super-category detection model (Farrell et al. 2011; Wah et al. 2011; Parkhi et al. 2011; Zhang et al. 2012; Liu et al. 2012; Parkhi et al. 2012) (*i.e.*, running a universal bird detector before a species classifier) implicitly use a form of part sharing. Similarly, many attribute-based methods (Lampert et al. 2009; Kumar et al. 2009; Farhadi et al. 2010) are motivated as a mechanism for model sharing.

An equally important motivation for parts and attributes is that they allow richer types of communication between humans and computers (Parikh and Grauman 2011; Parkash and Parikh 2012; Farhadi et al. 2009). In this paper, we aim to further develop this area, by introducing improved models and algorithms for human–computer-interaction based on parts and attributes.

3 Computer Vision for Fine-Grained Categories, Parts, and Attributes

3.1 Overview and Notation

In this section, we describe different flavors of computer vision algorithms that apply to multiclass recognition, part detection, and attribute detection. The computer vision algorithms described in this section obtain state-of-the-art performance on CUB-200-2011 (Wah et al. 2011) without any interactive component. As such, we describe them in this section as a standalone module, such that they maybe relevant to researchers who are working on a fully automatic solution to FGVC.

However, as the main point of this paper pertains to human-in-the-loop systems, we would also like to describe our algorithms in a way such that researchers who prefer different types of computer vision algorithms (*e.g.*, boosting instead of SVMs) could incorporate their algorithms in a human-in-the-loop framework. As such, we briefly introduce notation used throughout the paper and provide an overview

of how different types of computer vision algorithms can be mapped into an interactive framework.

The notation of this paper is a bit heavy, as we aim to combine computer vision and human estimates of classes, parts, and attributes. For all methods, we assume an image x belongs to a single class $c \in \{1...C\}$ (*i.e.*, each image contains a single bird species). For attribute-based methods, we assume an object can be represented by a vector of A attributes $\mathbf{a} = a_1...a_A$. For part-based methods, we assume an object's location can be represented by an array of P part locations $\Theta = \theta_1...\theta_P$. We use $\tilde{\mathbf{a}}$ and $\tilde{\Theta}$ to represent a human's perception of $\mathbf{a}$ and Θ, respectively. We use the notation $p_M(...)$ to indicate a probability estimated using machine vision, and $p_H(...)$ to indicate a probability estimated using human models. We begin by giving a high-level sketch of four types of computer vision algorithms and the basic methods for placing them in an interactive framework.

- *Multiclass classification techniques* (Sect. 3.2.1) are adapted to produce a probabilistic output $p_M(c|x)$. They are combined with human feedback by training a probabilistic model of how humans answer attribute questions $p_H(\tilde{a}_i|c)$ for each class separately.
- *Attribute-based recognition techniques* (Sect. 3.2.2) assume expert defined class-attribute memberships $\mathbf{a^c}$ (Lampert et al. 2009) and are adapted to produce a probabilistic output $p_M(c|x) \propto \prod_i p_M(a_i^c|x)$. They are combined with human feedback by training a probabilistic model of how humans answer attribute questions $p_H(\tilde{a}_i|a_i)$ for each type of attribute. By sharing attribute classifiers and human answer models among classes, they have potential to require fewer training images.
- *Localized multiclass methods* (Sect. 3.3) estimate class probabilities $p_M(c|x, \Theta)$ conditioned on a candidate detection location Θ, where Θ describes an object location and may encapsulate multiple parts or poses. They require adapting detection algorithms to produce a probabilistic output $p_M(\Theta|x)$. Part detectors may be shared among classes. A probabilistic model of how users answer part click questions $p_H(\tilde{\theta}_p|\theta_p)$ is used to refine part predictions.
- *Localized attribute-based methods* (Sect. 3.4) estimate $p_M(c|x, \Theta) \propto \prod_i p_M(a_i|x, \Theta)$ based on a set of attribute detectors. They integrate with humans via a models of $p_H(\tilde{\Theta}|\Theta)$ and $p_H(\tilde{a}_i|a_i)$.

3.2 Multiclass Recognition Without Localization

3.2.1 Multiclass Recognition

Many popular multiclass recognition methods such as SVMs, boosting, and logistic regression predict the class c with highest score: $\arg\max_c m^c(x)$. For example, in our implementation we assume a linear model

$$m^c(x) = \mathbf{w^c} \cdot \boldsymbol{\phi}(x) \tag{1}$$

where $\boldsymbol{\phi}(x)$ is a d dimensional feature vector and $\mathbf{w^c}$ is a d dimensional vector of learned weights. We learn $\mathbf{w} = \mathbf{w^1}, ..., \mathbf{w^C}$ jointly using a Crammer-Singer multiclass SVM

$$\min_{\mathbf{w},\epsilon} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{j=1}^{N}\epsilon_j \tag{2}$$

$$\text{s.t., } \forall_{j,c\neq y_j}, m^c(x_j) + 1 \leq m^{y_j}(x_j) + \epsilon_j \tag{3}$$

over a training set of N image-class pairs (x_i, y_i). This objective attempts to learn weights such that for each example x_i, the score of the true class $\langle \mathbf{w}^{y_i}, \boldsymbol{\phi}(x_i)\rangle$ is greater than the score of every other class $\langle \mathbf{w}^c, \boldsymbol{\phi}(x_i)\rangle$, incorporating a penalty via a slack variable ϵ_i when this is impossible. We convert scores $m^c(x)$ to probabilities using Platt scaling (Platt 1999), where probabilities are estimated using multiclass sigmoids, the parameters of which are chosen by maximizing the log-likelihood on a validation set of L images

$$p_M(c|x) = \frac{\exp\{\kappa^c m^c(x) + \delta^c\}}{\sum_{c'}\exp\{\kappa^{c'} m^{c'}(x) + \delta^{c'}\}} \tag{4}$$

$$\boldsymbol{\kappa}^*, \boldsymbol{\delta}^* = \arg\max_{\boldsymbol{\kappa},\boldsymbol{\delta}} \sum_{i=1}^{L} \log p_M(c_i|x_i) \tag{5}$$

In practice, we found that learning only a single parameter κ that is shared among classes worked just as well (possibly due to joint training of class weight vectors in Eq. 3). This simpler model results in probabilistic estimates

$$p_M(c|x) = \frac{\exp\{\kappa m^c(x)\}}{\sum_{c'}\exp\{\kappa m^{c'}(x)\}} \tag{6}$$

3.2.2 Attribute-Based Recognition

Different bird species are often composed of the same basic colors, patterns, and shapes (Farhadi et al. 2009; Lampert et al. 2009; Kumar et al. 2009; Farhadi et al. 2010; Wang and Forsyth 2009; Parikh and Grauman 2011). Exploiting these similarities offers the potential to learn from fewer training examples, and share processing between classes. As in (Lampert et al. 2009), we assume each class c is represented by an A-dimensional vector of binary[1] attributes $\mathbf{a^c} = a_1^c, ..., a_A^c$ (*e.g.*, a_i^c could indicate that a blue jay has a blue back). A weight vector $\mathbf{w_i^a}$ is learned for each attribute, producing a classification score $m_i^a(x) = \mathbf{w_i^a} \cdot \boldsymbol{\phi}(x)$. Let $\mathbf{m}^a(x) = m_1^a(x), ..., m_A^a(x)$ be a vector of attribute classification scores. In our experiments, we consider two possible

[1] Our user model assumes binary or multinomial attributes; however, one could use continuous attribute values for the computer vision component described in this section

ways of training class-attribute methods. In the first approach, we independently train a binary classifier for each attribute i, as in (Lampert et al. 2009):

$$\min_{\mathbf{w}_i^a,\epsilon} \frac{\lambda}{2}\|\mathbf{w}_i^a\|^2 + \frac{1}{N}\sum_{j=1}^{N}\epsilon_j, \quad \text{s.t., } \forall_j, 1 \le m_i^a(x_j) b_i^{y_j} + \epsilon_j \tag{7}$$

where $b_i^{y_j} = 2a_i^{y_j} - 1$. In our second approach, we learn attributes jointly while maximizing multiclass classification accuracy, optimizing Eq. 3 where class scores are computed as $m^c(x) = \mathbf{a^c} \cdot \mathbf{m}^a(x)$. This corresponds to the same probabilistic model and parameterization as the direct-attribute-model from (Lampert et al. 2009); however, whereas (Lampert et al. 2009) trains attributes independently and then uses a validation set to normalize them with respect to one another, we train attributes jointly and discriminatively with respect to a set of observed classes (*i.e.*, optimizing Eq. 3). Additional details for solving this convex optimization problem is contained in the supplementary material.

3.3 Multiclass Recognition With Localization

Reent work (Farrell et al. 2011; Wah et al. 2011; Parkhi et al. 2011; Zhang et al. 2012; Liu et al. 2012; Parkhi et al. 2012) has suggested that more strongly localized algorithms maybe necessary to solve FGVC problems. Let Θ be some encoding of the location of an object within an image; for example, it could encode information about the object's bounding box, part locations, pose or viewpoint, or 3D geometry. Class probabilities can be computed by marginalizing over part locations:

$$p_M(c|x) = \int p_M(c|x,\Theta)p_M(\Theta|x)d\Theta \tag{8}$$

Here, $p_M(\Theta|x)$ is the probability that an object is in a particular configuration Θ, and can be computed using techniques from object detection or part-based detection (Sect. 3.3.1). $p_M(c|x,\Theta)$ can be computed as the output of a localized multiclass classifier (Sect. 3.3.2) which extracts features with respect to a candidate configuration Θ [*i.e.*, in some pose normalized space (Zhang et al. 2012)]. In this paper, we represent the location of an object by a set of part locations $\Theta = \{\theta_1 ... \theta_P\}$, where the location $\theta_p = \{x_p, y_p, s_p, v_p\}$ of a particular part p is represented as an image location (x_p, y_p), a scale s_p, and an aspect v_p (*e.g.*, side view left, side view right, frontal view, not visible, *etc.*).

3.3.1 Part Detection

We represent parts using a deformable part model (DPM) (Felzenszwalb and Huttenlocher 2002), where parts are arranged in a tree-structured graph $T = (V, E)$ (see Fig. 6b). A full description of the model, inference, and learning is contained in the supplementary material. We review the basic terminology here. We model the detection score $g(\Theta; x)$ as a sum over unary and pairwise potentials $\log(p_M(\Theta|x)) \propto g(\Theta; x)$ with

$$g(\Theta; x) = \sum_{p=1}^{P}\psi(\theta_p; x) + \sum_{(p,q)\in E}\lambda(\theta_p, \theta_q) \tag{9}$$

where each unary potential $\psi(\theta_p; x)$ is the response of a sliding window part detector, and each pairwise score $\lambda(\theta_p, \theta_q)$ encodes a likelihood over the relative displacement between adjacent parts. We use the same learning algorithms and parameterization of each term in Eq. 9 as in (Branson et al. 2011; Yang and Ramanan 2011). Here parts are semantically defined, and weight parameters for appearance and spatial terms are learned jointly using a structured SVM (Tsochantaridis et al. 2006).

A mixture model is used to handle objects of different poses, such that the part detection score $\psi(\theta_p; x)$ is set equal to the detection score for the selected aspect (mixture component) v_p. The aspect v_p is latent during test time, but is assumed to be observed during structured SVM training. Since the datasets that we use label part locations (x_p, y_p) but not aspects v_p, aspect labels are assigned prior to training using pose clustering techniques—this practice is widely used for popular implementations of DPMs (Yang and Ramanan 2011) and poselets (Bourdev and Malik 2009). In the supplementary material, we consider two pose clustering techniques—one by clustering segmentation masks around labeled part locations and another by clustering offsets between pairs of parts. Mixture models are used to handle both variation in pose/viewpoint as well as variation due to species of different shape, since objects in certain poses or of certain species will be more likely to be assigned to the same aspect labels. At the same time, the same set of part-aspect detectors are shared among different species, yielding improved computuational properties and generalization.

After training, we convert detection scores to probabilities $p_M(\Theta|x) \propto \exp(\gamma g(\Theta; x))$, where γ is a scaling parameter that is learned by maximizing the log-likelihood on a validation set of L images labeled by ground truth parts. Let Θ_i denote the ground truth part labels of image x_i:

$$p_M(\Theta|x) = \frac{\exp(\gamma g(\Theta; x))}{\sum_\Theta \exp(\gamma g(\Theta; x))} \tag{10}$$

$$\boldsymbol{\gamma}^* = \arg\max_{\boldsymbol{\gamma}} \sum_{i=1}^{L} \log p_M(\Theta_i|x_i) \tag{11}$$

Note that although the denominator of Eq. 10 occurs over an exponentially large set of part locations, it can be computed in time linear in the number of parts using dynamic programming. Examples of fully automated part detection results are shown in Fig. 4.

3.3.2 Localized Multiclass Recognition

We include more details about how to adapt multiclass and attribute based recognition techniques with a localization model in the supplementary material; however, the basic idea is that for each detected part location θ_p, features $\boldsymbol{\varphi}_{\mathbf{p}}(\theta_p; x)$ are extracted from some localized region of interest around θ_p (see Fig. 5). Features for each part $p = 1...P$ can be concatenated into one long feature vector

$$\boldsymbol{\Psi}(\Theta; x) = [\boldsymbol{\varphi}_{\mathbf{1}}(\theta_1; x), ..., \boldsymbol{\varphi}_{\mathbf{P}}(\theta_P; x)] \tag{12}$$

$$\mathbf{w}^{\mathbf{c}} = [\mathbf{w}^{\mathbf{c}}_{\mathbf{1}}, ..., \mathbf{w}^{\mathbf{c}}_{\mathbf{P}}] \tag{13}$$

The feature space $\boldsymbol{\Psi}(\Theta; x)$ is a pose-normalized feature space that is extracted with respect to a candidate set of part configurations Θ. If we know that an object is in a configuration Θ, a multiclass classification can be performed as:

$$m^c(\Theta; x) = \mathbf{w}^{\mathbf{c}} \cdot \boldsymbol{\Psi}(\Theta; x) = \sum_p \mathbf{w}^{\mathbf{c}}_{\mathbf{p}} \cdot \boldsymbol{\varphi}_{\mathbf{p}}(\theta_p; x) \tag{14}$$

At train time, we assume that each training image x_i has been labeled with ground truth part locations Θ_i and a class label y_i. We learn weights $\mathbf{w}^{\mathbf{c}}$ using a multiclass SVM (Eq. 3) using features extracted at ground truth part locations, $\phi(x_i) = \boldsymbol{\Psi}(\Theta_i; x_i)$. We produce probabilistic estimates $p_M(c|x, \Theta)$ using Eq. 6 and Eq. 14.

3.4 Attribute-Based Recognition with Localization

A similar approach can be used to augment the attribute-based model described in Sect. 3.2.2 with a part localization model. Here, attribute detection scores

$$m^a_i(\Theta; x) = \mathbf{w}^{\mathbf{a}}_{\mathbf{i}} \cdot \boldsymbol{\Psi}(\Theta; x) \tag{15}$$

for each attribute i are combined into a vector $\mathbf{m}^a(\Theta; x)$, which induces multiclass classification scores

$$m^c(\Theta; x) = \mathbf{a}^{\mathbf{c}} \cdot \mathbf{m}^a(\Theta; x) \tag{16}$$

A combined model that learns both per-class weights and attribute weights that are shared among classes is also possible:

$$m^c(\Theta; x) = \mathbf{a}^{\mathbf{c}} \cdot \mathbf{m}^a(\Theta; x) + \mathbf{w}^{\mathbf{c}} \cdot \boldsymbol{\Psi}(\Theta; x) \tag{17}$$

The motivation for using this model is that the per-class model (Eq. 14) usually tends to outperform the attribute-based model (Eq. 16) in practice–presumably because an A-dimensional attribute-space is too simple to discriminate classes well—(see Sect. 6), whereas the shared attribute model can improve generalization when the number of training examples per class is small.

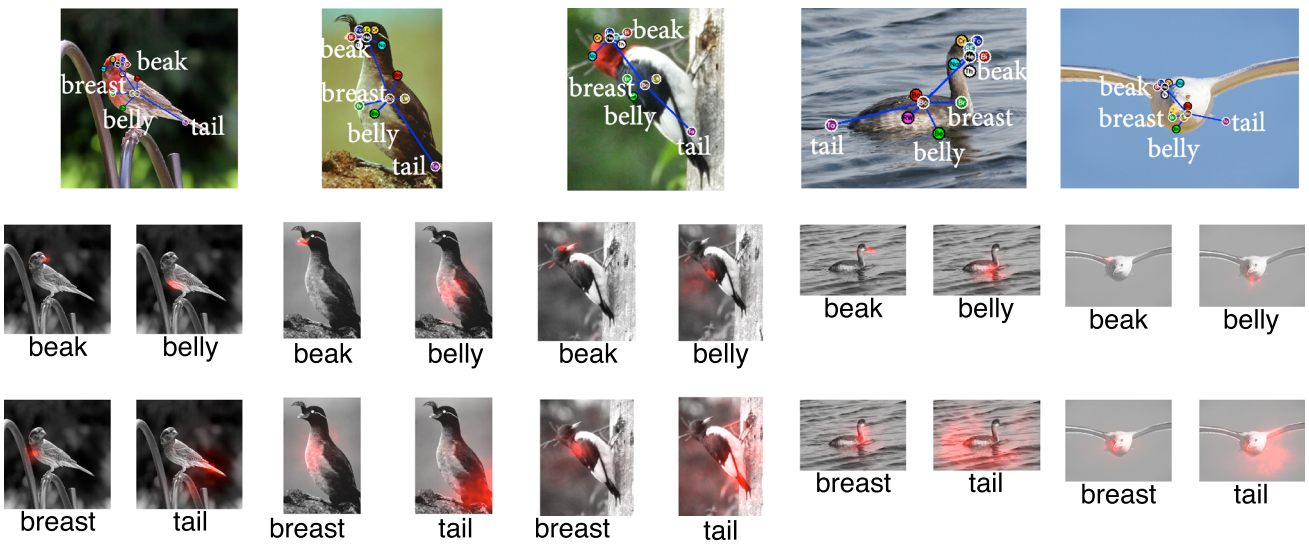


Fig. 4 Fully automated part detection results. five test images with maximum likelihood estimates of 15 semantic parts superimposed on the image, and marginalized part probability maps for four parts. Our system does a good job localizing all parts for the first two images, as is typical with side and frontal views of birds. The 3rd image is in an unusual horizontal pose; our system detects the parts of the head correctly but flips the orientation of the body upside down. The 4th image is an unusual bird *shape*; our system detects all parts more or less correctly but with some degree of noise. The last image is an uncommon pose for which detection fails entirely

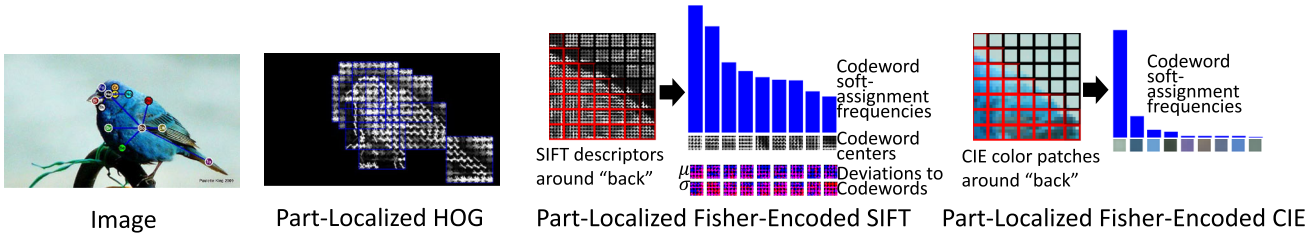


Fig. 5 Visualization of features for localized multiclass classification. Three types of features are extracted around detected part locations in the bird image on the left. From *left* to *right* (1) 7×7 HOG templates (used for detection), (2) SIFT descriptors are extracted from patches around a part location and weighted by a ROI predictor based on the average segmentation mask for the predicted aspect label (weights visualized in *red*). Patches are soft-assigned to a codebook, and used to induce a Fisher vector feature space (Perronnin et al. 2010), which can be interpreted as a higher-dimensional version of bag-of-words that encodes the deviations of patches to assigned codewords w.r.t. each SIFT descriptor dimension. (3) The same procedure is used on raw patches in CIE-Lab-color space instead of on SIFT descriptors (Color figure online)

4 Human Recognition of Fine-Grained Categories, Parts, and Attributes

In the previous section, we described computer vision algorithms that produce probabilistic outputs for predictions of classes $p_M(c|x)$, attributes $p_M(a_i|x)$, part locations $p_M(\Theta|x)$, and localized class probabilities $p_M(c|x,\Theta)$. Recall from Sect. 3.1, that these can be combined with human-interactive algorithms if one can train models of how humans answer attribute questions $p_H(\tilde{a}_i|c)$ and perceive object or part locations $p_H(\tilde{\Theta}|\Theta)$. In this section, we introduce ways of modeling these two types of probabilities and then estimate their parameters using experiments on Mechanical Turk. Our experiments are conducted using the CUB-200-2011 dataset (Wah et al. 2011).

While we have not performed scientific studies of human perception of fine-grained visual categories, it should be clear that the recognition performance of non-experts is extremely low (*e.g.*, the average person has not heard of a *Pied-billed Grebe* and therefore cannot identify it). By contrast, in a small scale experiment we found that expert birders could achieve around 93 % accuracy on CUB-200-2011; the number is less than 100 % because other cues such as multiple views of the bird, sounds, behavior, and geographical location are often necessary for accurate recognition.

4.1 Attributes

We constructed a set of 312 binary-valued visual attributes over 29 attribute groupings (*e.g.*, the grouping *bird shape* has 14 different possible shapes such as *gull-like*, *duck-like*, *etc.*). The attributes were derived from the birding website www.whatbird.com.

4.1.1 Binary Questions

Let a_i be the ground truth value of a binary attribute (*e.g.*, *is the belly white?*) and $\tilde{a}_i$ be a random variable for a user's perception of a_i. We model probabilities for each class $\hat{p}_i^c = p_H(\tilde{a}_i|c)$ as a simple binomial distribution with a Beta prior $\mathrm{B}(\beta\hat{p}_i, \beta\hat{q}_i)$, where β is a constant, $\hat{p}_i = p_H(\tilde{a}_i)$ is a global attribute prior, and $\hat{q}_i = 1 - \hat{p}_i$. Suppose we have a training set $(x_1, c_1, \tilde{\mathbf{a}}^{\mathbf{1}}), ..., (x_n, c_n, \tilde{\mathbf{a}}^{\mathbf{n}})$, where each image x_j is labelled by a class c_j and attribute responses $\tilde{\mathbf{a}}^{\mathbf{j}} = \tilde{a}_1^j, ..., \tilde{a}_A^j$. Then the MAP estimate of $\hat{p}_i^c$ is:

$$\hat{p}_i = \frac{\sum_j \tilde{a}_i^j}{n} \tag{18}$$

$$\hat{p}_i^c = \frac{\beta\hat{p}_i + \sum_j \tilde{a}_i^j 1[c_j = c]}{\beta + \sum_j 1[c_j = c]} \tag{19}$$

where 1[...] denotes the indicator function. In other words, $p_H(\tilde{a}_i)$ is estimated as the fraction of the time MTurkers answer yes to the ith attribute question irrespective of class, whereas $p_H(\tilde{a}_i|c)$ is estimated as the fraction of time MTurkers answer yes for the ith attribute for images of class c if we add β synthetic examples from the distribution of $p_H(\tilde{a}_i)$.

4.1.2 Multiple Choice Questions

Attributes within some attribute groupings such as *bird shape* are assumed to be mutually exclusive (*i.e.*, bird shape is a multiple choice question). The extension to these types of questions is straightforward. Here, we assume a response $\tilde{a}_i$ has k_i possible discrete values $a_i \in 1...k_i$. We model $p_H(\tilde{a}_i|c)$ as a Multinomial distribution with a Dirichlet prior, resulting in estimates:

$$\hat{p}_{ik} = \frac{\sum_j 1[\tilde{a}_i^j = k]}{n} \tag{20}$$

$$\hat{p}_{ik}^c = \frac{\beta\hat{p}_{ik} + \sum_j 1[\tilde{a}_i^j = k, c_j = c]}{\beta + \sum_j 1[c_j = c]} \tag{21}$$

Here $\hat{p}_{ik} = p_H(\tilde{a}_i = k)$ is computed as the fraction of the time MTurkers choose value k for the ith attribute question

irrespective of class, whereas $\hat{p}^c_{ik} = p_H(\tilde{a}_i = k|c)$ is computed as the fraction of time MTurkers choose value k for the ith attribute question for images of class c if we add β synthetic examples from the distribution of $p_H(\tilde{a}_i = k)$.

4.1.3 Per-Class Attribute Model Versus Per-Attribute Model

So far, we have assumed that attribute response probabilities are estimated separately for each class $p_H(\tilde{a}_i|c)$. An alternative is to train a model $p_H(\tilde{a}_i|a_i)$ for each possible value of a_i separately while ignoring class (*i.e.*, using a similar method as Eqs. 19 and 21). Note that estimating $p_H(\tilde{a}_i|c)$ requires human experiments that pose attribute questions for all possible class-attribute pairs (c, i), an operation that maybe expensive. By contrast, the corresponding attribute-based method trains models $p_H(\tilde{a}_i|a_i)$ for each attribute a_i, independent of class. Per-class answer probabilities can then be estimated as $p_H(\tilde{a}_i|c) = p_H(\tilde{a}_i|a^c_i)$, assuming expert-defined class-attribute values a^c_i are available (*e.g.*, from a field guide one can infer that the crown of a blue jay is blue).

On the positive side, this allows the possibility of introducing new unseen classes [analagous to (Lampert et al. 2009)] without requiring additional human experiments (*e.g.*, if we introduce a species *blue jay*, we can assume the distribution of how users answer the question *is the crown blue?* can be derived based on statistics of other observed bird species that have a blue crown). On the negative side, some information is lost in the mapping to expert-defined binary attributes a^c_i. Using a per-class model $p_H(\tilde{a}_i|c)$ will usually give better results if enough training data is available.

4.2 Modeling User Click Responses

In this section, we construct a model of human responses to the simple interface shown in Figs. 6, 7b, where the user is asked to click on the location of a part p, or specify that p is not visible. We represent a user's click response as a triplet $\tilde{\theta}_p = \{\tilde{x}_p, \tilde{y}_p, \tilde{v}_p\}$, where $(\tilde{x}_p, \tilde{y}_p)$ is a point that the user clicks with the mouse and $\tilde{v}_p \in \{0, 1\}$ is a binary variable indicating *not visible* or *visible* respectively.

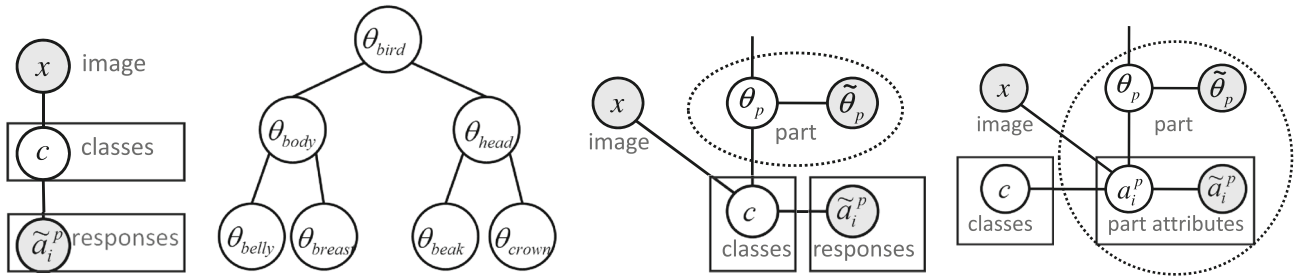
Note that the user click response $\tilde{\theta}_p$ models only part location and visibility, whereas our model of the part's true location $\theta_p = \{x_p, y_p, s_p, v_p\}$ also includes scale and aspect. This is done in order to keep the user interface as intuitive as possible. On the other hand, incorporating scale and aspect in the computer vision model is extremely important — the relative offsets and visibility of parts in *left side view* and *right side view* will be dramatically different. Let us assume that $v_p = 0$ indicates that part p is not visible and other values stand for different visible aspects. We assume that the user correctly predicts a part's visibility with some probability depending on the ground truth pose, modeling $p(\tilde{v}_p = 1|v_p)$ as a separate binomial distribution for each possible value of v_p. If the user correctly predicts visiblity and clicks somewhere, we assume the user's click location (normalized by the scale of the object) is Gaussian distributed from the ground truth location

$$\tilde{c}_p = \left(\frac{\tilde{x}_p - x_p}{s_p}, \frac{\tilde{y}_p - y_p}{s_p}\right), \qquad \tilde{c}_p \sim \mathcal{N}(\tilde{\mu}_p, \tilde{\Sigma}_p) \tag{22}$$

If the user incorrectly predicts that a part is visible, we assume that the user's click location is uniformly distributed throughout the image. Enumerating each of these cases:

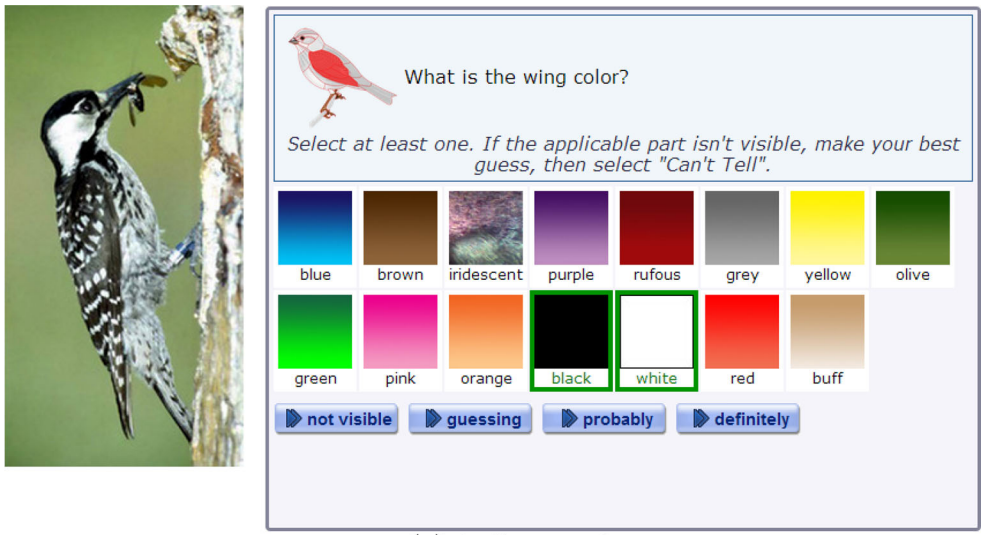
$$p_H(\tilde{\theta}_p|\theta_p) = p(\tilde{v}_p|v_p) \begin{cases} p_H(\tilde{c}_p|\tilde{\mu}_p, \tilde{\Sigma}_p) & \text{if } v_p \neq 0, \tilde{v}_p \neq 0 \\ 1 & \text{if } \tilde{v}_p = 0 \\ \frac{1}{WH} & \text{if } v_p = 0, \tilde{v}_p \neq 0 \end{cases} \tag{23}$$

where W and H are the width and height of the image, and $p_H(\tilde{c}_p|\tilde{\mu}_p, \tilde{\Sigma}_p)$ is the bivariate normal probability density with mean $\tilde{\mu}_p$ and covariance $\tilde{\Sigma}_p$. The parameters of



(a) Class Model **(b)** Part Model **(c)** Localized Class Model **(d)** Part-Attribute Model

Fig. 6 Probabilistic models. In this paper, we describe several different flavors of computer vision algorithms and how they can be combined with interactive feedback. (**a**) The unlocalized multiclass model (Sect. 5.1.1) trains a classifier and model of how users answer questions for each class independently. (**b**) Our localization model assumes spatial relationship between parts has a hierarchical independence structure. (**c**) A localized per-class model Sect. 5.1.2 incorporates the part-tree from (**b**) and trains a detector for each class. (**d**) A localized part-attribute model (Sect. 5.1.3) incorporates the part-tree from (**b**) and shares attribute detectors between classes



(a) Attribute question

(b) Part question

Fig. 7 Attribute and part questions. **a** For the attribute question *what is the wing color* the user selects both *black* and *white* and qualifies her answer with a certainty *definitely*. **b** For the part click question *click on the tail*, the user provides an (x, y) mouse location (Color figure online)

these distributions are estimated using a training set of pairs $(\theta_p, \tilde{\theta}_p)$. Figures 8 and 9b visualizes one standard deviation when we learned our model (Eq. 22) from over 26,000 clicks per part from Mechanical Turk workers. As a reference, we also include a comparison to computer vision part predictions (Sect. 3.3.1) in Fig. 9c.

5 Combining Humans and Computers

In Sect. 3, we described computer vision algorithms that produce probabilistic outputs for predictions of classes $p_M(c|x)$, attributes $p_M(a_i|x)$, part locations $p_M(\Theta|x)$, and localized class probabilities $p_M(c|x, \Theta)$. In Sect. 4, we introduced probabilistic models of how humans predict attributes $p_H(\tilde{a}_i|c)$ and part locations $p_H(\tilde{\Theta}|\Theta)$. In this section, we address the questions (1) how do we combine these different sources of information into an improved estimate that is better than humans or computers could do in isolation?, and (2) if we treat human time as a precious resource, how do we use our current beliefs to intelligently select what type of human input to query next?

We begin by describing our methods of combining computer vision and human responses into an improved estimate $p(c|x, U)$ in Sect. 5.1, where U is assumed to be a collection of user responses that we have received so far. In Sect. 5.2, we describe an active testing (Geman and Jedynak 1993, 1996) algorithm called the *Visual 20 Questions Game* (visualized in Fig. 10), in which a machine intelligently chooses questions to pose to a human user with the objective of identifying the true class as quickly and as accurately as possible. This interactive algorithm incorporates methods for estimating $p(c|x, U)$ as a sub-routine.

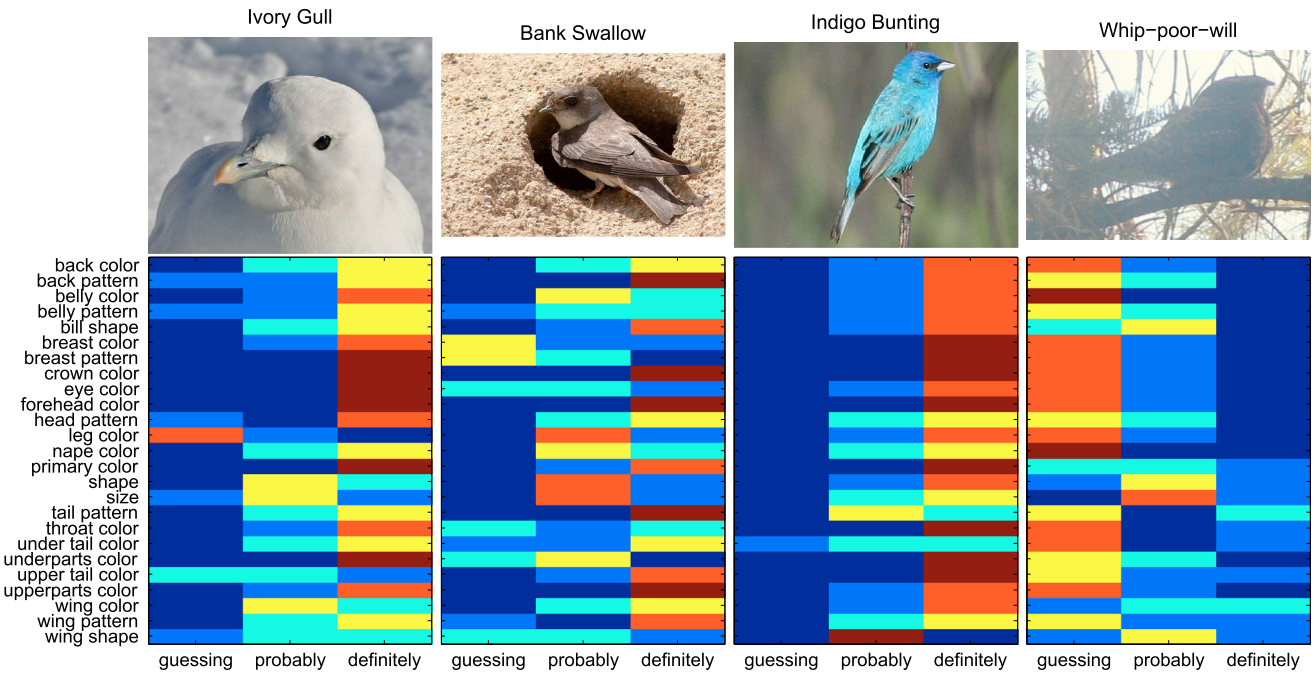


Fig. 8 Examples of user responses for 25 attribute groupings. The distribution over {*Guessing, Probably, Definitely*} is color coded with blue denoting 0 % and red denoting 100 % of the five answers per image attribute pair. Notice, for example, that the ivory gull image on the left receives unambiguous answers for the crown color and the eye color, while it receives highly uncertain answers for the color of the leg. Also, the Whip-poor-will image on the right is of bad quality and received many 'guessing' answers as a result

(a) Image-Level MTurk **(b)** Global MTurk **(c)** Computer Vision

Fig. 9 Comparing part prediction accuracy for humans and computers. In each case, a Gaussian distribution over scale-normalized offsets between predictions and ground truth is estimated (Eq. 23), and ellipses visualize 1 standard deviation from ground truth. (**a**) Image-level standard deviations over 5 MTurk users who labeled this particular Black-footed Albatross image. (**b**) Global standard deviations over 5,794 images and five users per image. Ellipses are superimposed onto an unrelated picture of a bird for visualization purposes. Global standard deviations appear larger than image-level ones because occasionally MTurkers click entirely on the wrong part. (**c**) Standard deviations over computer vision predictions (Sect. 3.3.1) for 5,794 test images. Standard deviations of computer vision predictions are much larger because occasionally computer vision detects the bird entirely in the wrong location

5.1 Combining Human and Machine Predictions

As it is our goal to make our formulation as general as possible, we break down different ways of estimating $p(c|x, U)$ into sections, each of which is applicable to a different family of computer vision algorithms. Section 5.1.1 pertains to traditional unlocalized multi-class classification algorithms. Section 5.1.2 incorporates part localization and a localized classification model. Section 5.1.3 extends this model further by sharing attribute detectors between classes. These three methods correspond to the probabilistic models shown in Fig. 6a, 6c, and 6d respectively. All three methods support human interaction via human attribute responses, whereas interaction via part click questions pertains only to the localized models.

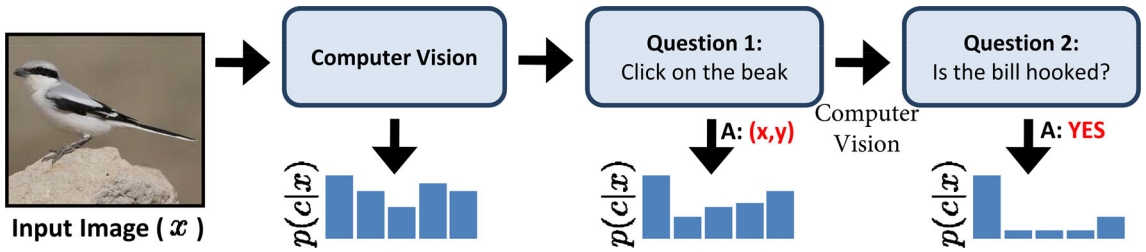


Fig. 10 Visualization of the flow of the basic algorithm. The system poses questions to the user, which along with computer vision, incrementally refine the probability distribution over classes

5.1.1 Combining Multiclass Recognition with Human Attribute Responses

In this section, we propose a simple method for combining traditional multiclass recognition algorithms (see Sect. 3.2.1) with answers to human attribute questions $U = \tilde{\mathbf{a}}$ (see Sect. 4.1). Our discussion in this section pertains to unlocalized computer vision algorithms–assuming the simple model depicted in Fig. 6a—such that part click questions are not relevant. The probability $p(c|x, \tilde{a})$ can be written as:

$$p(c|x,\tilde{a}) = \frac{p(c,\tilde{a}|x)}{p(\tilde{a}|x)} = \frac{p(\tilde{a}|c,x)p(c|x)}{\sum_{c'} p(\tilde{a}|c',x)p(c'|x)} \tag{24}$$

We define these two expressions in terms of our computer vision model $p(c|x) = p_M(c|x)$—defined in Sect. 3.2.1—and human model $p(\tilde{a}|c, x) = p_H(\tilde{a}|c)$—defined in Sect. 4.1. In the latter case, we have modelled the user's perception of attributes as depending only on the class c of an object and not the image x. This is reasonable if we assume that attributes are class-deterministic, and the human brain is able to parse an image into detected attributes while factoring out other external factors contained within x such as pose and lighting. Note that our model of $p_H(\tilde{\mathbf{a}}|c)$ is still non-deterministic, allowing us to accommodate for variation in responses due to user error, subjectivity in naming attributes (*e.g.*, different people perceive the color *blue* differently), and other sources of intraclass variance.

5.1.2 Localized Multiclass Model

In this section, we describe an extension to the algorithms described in the previous section to localized computer vision algorithms (see Sect. 3.3.1), where one first attempts to predict both the location of an object in an image (*e.g.*, the location and pose of different parts) as well as its class. Such algorithms offer additional opportunities for a more complex interplay between computer vision algorithms and human interactivity, because a person can provide interactive feedback with respect to her perception of both object localization as well as object attributes.

Incorporating a localization model, the class probabilities can be obtained by marginalizing over the localization variables Θ:

$$p(c|x,U) = \frac{p(c,U|x)}{\sum_c p(c,U|x)} \tag{25}$$

$$p(c,U|x) = \int_\Theta p(c,U,\Theta|x)d\Theta \tag{26}$$

Note that $p(c, U, \Theta|x)$ can be decomposed into terms

$$p(c,U,\Theta|x) = p(c|\Theta,x)p(\Theta|x)p(U|c,\Theta,x) \tag{27}$$

where we define $p(c|\Theta, x) = p_M(c|\Theta, x)$ in terms of the output of a localized multiclass classifier (see Sect. 3.3.2), and $p(\Theta|x) = p_M(\Theta|x)$ in terms of the output of a part-based detector (see Sect. 3.3.1). Suppose we separate U into sets $U_\Theta \subseteq U$ and $U_a \subseteq U$ that pertain to part and attribute responses respectively. We define $p(U|c, \Theta, x)$ in terms of our user models developed in Sect. 4

$$p(U|c,\Theta,x) = p_H(U_\Theta|\Theta)p_H(U_a|c) \tag{28}$$

$$= \left(\prod_{\tilde{\theta}_p \in U_\Theta} p_H(\tilde{\theta}_p|\theta_p)\right)\left(\prod_{\tilde{a}_i \in U_a} p_H(\tilde{a}_i|c)\right) \tag{29}$$

Here, we have applied the independence assumptions depicted in Fig. 6c; we assume a user's perception of the location of a part p depends only on the ground truth location of that part $p(\tilde{\theta}_p|\theta_p)$ (see Sect. 4.2), and a user's perception of an attribute a_i depends only on the ground truth class, as justified in Sect. 5.1.1.

5.1.3 Localized Attribute Model

A related model uses attribute-based detection in place of standard multiclass classification techniques:

$$\begin{aligned} p(c,U,\Theta|x) &= \sum_{\mathbf{a}} p(c,U,\Theta,\mathbf{a}|x) \\ &= \sum_{\mathbf{a}} p_M(c,\mathbf{a}|\Theta,x)p_M(\Theta|x)p_H(U|c,\mathbf{a},\Theta,x) \\ &= p_M(\mathbf{a}^\mathbf{c}|\Theta,x)p_M(\Theta|x)p_H(U_\Theta|\Theta)p_H(U_a|\mathbf{a}^\mathbf{c}) \end{aligned} \tag{30}$$

where we assume each class c deterministically has a unique vector of attributes $\mathbf{a^c}$ (Lampert et al. 2009) (see Fig. 6d), and $p(\mathbf{a}^c|\Theta, x)$ is the response of a set of attribute detectors evaluated at locations Θ (see Sect. 3.3.2). Note that in comparison to Eq. 29, we use a slightly different expression for the probability of human attribute responses $p(U_a|c)$:

$$p_H(U_a|\mathbf{a^c}) = \prod_{\tilde{a}_i \in U_a} p_H(\tilde{a}_i|a_i^c) \tag{31}$$

where we have incorporated per-attribute user models instead of per-class attribute user models (see Sect. 4.1.3).

5.1.4 Inference

In this section, we describe efficient inference procedures for estimating per-class probabilities $p(c|U, x)$ (Eq. 26) (either according to the localized class model in Fig. 6a or the localized part-attribute model Fig. 6c), which involves evaluating $\int_\Theta p(c, U, \Theta|x)d\Theta$. We note that all user responses $\tilde{a}_p^i$ and $\tilde{\theta}_p$ are observed values pertaining only to a single part, and attributes $\mathbf{a^c}$ are deterministic when conditioned on a particular choice of class c. If we run inference separately for each class c, the output of class detectors, part detectors, and user responses can all be combined and mapped into a unary potential for each part

$$\psi_p^c(\theta_p; x) = \kappa \mathbf{w_p^c} \cdot \boldsymbol{\varphi}_\mathbf{p}(\theta_p; x) + \gamma\psi(\theta_p; x) + \log p(\tilde{\theta}_p|\theta_p) \tag{32}$$

such that $g_U^c(\Theta; x) = \log p(c, U, \Theta|x)$ is expressible in canonical form for pictorial structure problems

$$g_U^c(\Theta; x) = K_U^c + \sum_{p=1}^{P} \psi_p^c(\theta_p; x) + \sum_{(p,q)\in E} \gamma\lambda(\theta_p, \theta_q) \tag{33}$$

where $K_U^c = \sum_{\tilde{a}_i \in U_A} \log p(\tilde{a}_i|c)$. The above expression can be obtained by plugging in the expressions from Eqs. 29, 14, 10, and 23 into Eq. 27. Thus evaluating Eq. 26 exactly can be done by running a separate deformable part model inference problem for each class[2].

On the other hand, when C is large, running C inference problems can be inefficient. In practice, we use a faster procedure that approximates the integral in Eq. 26 as a sum over K strategically chosen sample points:

$$\begin{aligned}\int_\Theta p(c, U, \Theta|x)d\Theta &\approx \sum_{k=1}^{K} p(c, U, \Theta^k|x) \\ &= \sum_{k=1}^{K} p_H(U|c, \Theta^k, x)p_M(c|\Theta^k, x)p_M(\Theta^k|x) \\ &= p_H(U_a|c)\sum_{k=1}^{K} p_M(c|\Theta^k, x)p_H(U_\Theta|\Theta^k, x)p_M(\Theta^k|x)\end{aligned} \tag{34}$$

We select the sample set $\Theta^1 \ldots \Theta^K$ as the set of all local maxima in the probability distribution $p(U_\Theta|\Theta)p(\Theta|x)$, where $f_U(\Theta; x) = \log(p(U_\Theta|\Theta)p(\Theta|x))$ is expressible as a pictorial structure problem with part detection and click likelihood scores combined into a unary potential $\tilde{\psi}_p(x, \tilde{\theta}_p; \Theta)$

$$f_U(\Theta; X) = \sum_{p=1}^{P} \tilde{\psi}_p(\theta_p; \tilde{\theta}_p, x) + \sum_{(p,q)\in E} \gamma\lambda(\theta_p, \theta_q) \tag{35}$$

$$\tilde{\psi}_p(\theta_p; \tilde{\theta}_p, x) = \gamma\psi(\theta_p; x) + \log p(\tilde{\theta}_p|\theta_p) \tag{36}$$

The set of local maxima and their respective log probabilities can be found using standard methods for maximum likelihood inference on deformable part models and then running non-maximal suppression. The inference step takes time linear in the number of parts and pixel locations[3] and is efficient enough to run in a fraction of a second with 15 parts, 30 aspects per part, and 4 scales. Inference is re-run each time we obtain a new user click response $\tilde{\theta}_p$, resulting in a new set of samples. Sampling assignments to part locations ensures that localized multiclass classification algorithms only have to be evaluated on K candidate assignments to part locations; this opens the door for more expensive categorization algorithms (such as kernelized methods) that do not have to be run in a sliding window fashion.

5.2 The Visual 20 Questions Game

In this section, we describe an interactive classification method called the visual 20 questions game that combines the models and algorithms that we have heretofore described in this paper. The algorithm is conceptually simple and summarized in Fig. 10; it poses a series of questions to a human user that are intelligently selected based on computer vision and previous user responses.

Let $\mathcal{Q} = \{q_1 \ldots q_n\}$ be a set of possible questions (*e.g.*, *is red*?, *has stripes*?, *click on the beak*, *etc.*), and $\mathcal{A}_i$ be the

[2] The integral in Eq. 26 involves a bottom-up traversal of $T = (V, E)$, at each step convolving a spatial score map with a unary score map (takes time $O(n \log n)$ time in the number of pixels).

[3] Maximum likelihood inference involves a bottom-up traversal of T, doing a distance transform operation (Felzenszwalb et al. 2008) for each part in the tree (takes time $O(n)$ time in the number of pixels).

Algorithm 1 Visual 20 Questions Game

1: $U^0 \leftarrow \emptyset$
2: **for** $t = 1$ to 20 **do**
3: $j(t) = \max_k \text{IG}(c; u_k|x, U^{t-1})$
4: Ask user question $q_{j(t)}$, and $U^t \leftarrow U^{t-1} \cup u_{j(t)}$.
5: **end for**
6: Return class $c^* = \max_c p(c|x, U^t)$

set of possible answers to q_i. The user's answer is some random variable $u_i \in \mathcal{A}_i$. At each time step t, we select a question $q_{j(t)}$ to pose to the user, where $j(t) \in 1...n$. Let $j \in \{1...n\}^T$ be an array of T indices to questions that we will ask the user. $U^{t-1} = \{u_{j(1)}...u_{j(t-1)}\}$ is the set of responses obtained by time step $t-1$. For our basic algorithm, we use maximum expected information gain as the criterion to select $q_{j(t)}$. We propose a different criterion based on minimizing expected human time in Sect. 5.2.4. Information gain is widely used in decision trees [*e.g.* (Quinlan 1993)] and can be computed from an estimate of $p(c|x, U^{t-1})$. (Geman and Jedynak 1993, 1996) introduced a "20-Questions-Game" approach for recognition that successively chooses a question to ask by computing information gain in online fashion (rather than precomputing an intractibly big decision tree). Our approach is an instance of this framework where the prediction model combines information from humans and computers. The expected information gain $\text{IG}(c; u_i|x, U^{t-1})$ of posing the additional question q_i, is defined as follows:

$$\text{IG}(c; u_i|x, U^{t-1}) = \sum_{u_i \in \mathcal{A}_i} p(u_i|x, U^{t-1})\big(\text{H}(c|x, U^{t-1}) - \text{H}(c|x, u_i \cup U^{t-1})\big) \tag{37}$$

where $p(u_i|x, U^{t-1})$ is an estimated probability that the user will answer u_i to the question q_i and $\text{H}(c|x, U^{t-1})$ is the entropy of $p(c|x, U^{t-1})$

$$\text{H}(c|x, U^{t-1}) = -\sum_{c=1}^{C} p(c|x, U^{t-1}) \log p(c|x, U^{t-1}) \tag{38}$$

The general algorithm for interactive object recognition is shown in Algorithm 1. Recall that we have already introduced methods for estimating $p(c|x, U)$, the main term in the entropy computation, in the previous section. In the remainder of this section, we describe techniques for efficiently solving $\max_i \text{IG}(c; u_i|x, U^{t-1})$ for several different flavors of computer vision algorithms and sources of user input.

5.2.1 Binary and Multiple Choice Attribute Questions

We first consider simple binary and multiple choice questions. These allow for a particularly simple online method for computation of $p(c|x, U^t)$ and $p(u_i|x, U^{t-1})$, the two terms in Eq. 37. Let us define $s^c_{t-1} = p_H(U^{t-1}|c)p_M(c|x)$ as the numerator of Eq. 24 after the $(t-1)th$ question. Note that $s^c_0 = p_M(c|x)$ can be precomputed using the computer vision algorithms defined in Sect. 3.2.1 or 3.3. Suppose we have already computed s^c_{t-1} in an earlier timestep and want to estimate an updated probability $p(c|x, U^{t-1}, \tilde{a}_j)$ after an additional user response $\tilde{a}_j$. If we use the model defined in Sect. 5.1.1, it follows that

$$p(c, U^{t-1}, \tilde{a}_j|x) = s^c_t = \hat{p}^c_j s^c_{t-1} \tag{39}$$

while the probability that the user will answer $u = \tilde{a}_j$ is

$$p(u|x, U^{t-1}) = \frac{\sum_c \hat{p}^c_j s^c_{t-1}}{\sum_{c, \tilde{a}'_j \in \mathcal{A}_q} \hat{p}^c_{j'} s^c_{t-1}} \tag{40}$$

and the resulting updated class probabilities are

$$p(c|x, U^{t-1}, \tilde{a}_j) = \frac{\hat{p}^c_j s^c_{t-1}}{\sum_{c'} \hat{p}^{c'}_j s^{c'}_{t-1}} \tag{41}$$

Eq. 37–41 define an efficient way for computing the expected information gain (Eq. 37) of a candidate question q.

5.2.2 Multi-Select and Batch Questions

We define batch questions as a collection of multiple questions that are more efficient for the user to answer at the same time than to answer sequentially. For example, as shown in Fig. 7a the question *what is the wing color* has 15 possible color choices, and the user can select more than one (in this case she selected *black* and *white*). As such, the question is similar to asking 15 simultaneous binary questions. We model this type of question q as a collection of L_q sub-questions $d_{q1}, ..., d_{qL_q}$. This poses a challenge when computing expected information gain, as the space of possible answers that we must search through $\mathcal{A}_q = \mathcal{A}_{q1} \times \mathcal{A}_{q2} \times ... \times \mathcal{A}_{qL_q}$ is exponential in the number of sub-questions.

We consider an approximation, where we instead search over a smaller set of K random samples $\bar{\mathcal{A}}_q = \bar{u}_1, ..., \bar{u}_K$, with each $\bar{u}_i$ defining an answer to all sub-questions. In practice, we draw each sample by looping over each sub-question d_{qk} and randomly choosing an answer according to its probability (Eq. 40). The probabilities $p(c, U^{t-1}, \tilde{a}_j|x)$ can then be estimated as in Sect. 5.2.1. Although this procedure is clearly sub-optimal (both due to sampling and treating sub-questions as independent), it is more important to have a fast question selection method (*i.e.*, never forcing the user to wait for the machine to process) than to choose the absolute optimal question (since typically many questions will provide useful information).

5.2.3 Part Click Questions

Part click questions (see Fig. 7b) pose an even more significant computational challenge, both because the number of possible answers to each question is large (equal to the number of pixel locations), and because the effect of each answer is complex (it involves refining estimates of part locations and integrating over them to recompute class probabilities). Evaluating the expected information gain (Eq. 37) for a given part location question q_j involves computing the expected entropy:

$$\mathbb{E}_{\tilde{\theta}_p}[\mathrm{H}(c|x, U^{t-1}, \tilde{\theta}_p)] = \sum_{\tilde{\theta}_p} p(\tilde{\theta}_p|x, U^{t-1})\, \mathrm{H}(c|x, U^{t-1}, \tilde{\theta}_p) \quad (42)$$

Using the model defined in Sect. 5.1.2, $p(\tilde{\theta}_p|x, U^{t-1})$ can be efficiently computed without approximation densely for all values of $\tilde{\theta}_p$ using dynamic programming (as a deformable part model inference problem), where our model of part clicks $\log p_H(\tilde{\theta}_p|\theta_p)$ has been mapped into a pairwise potential between nodes $\tilde{\theta}_p$ and θ_p. Note that this is possible because adding unobserved variables $\tilde{\theta}_p$ to the tree-structured graphical model depicted in Fig. 6b preserves a tree-structured graph. In practice, computing probabilities $p(\tilde{\theta}_p|x, U^{t-1})$ for all values of p while marginalizing over Θ can be computed using a single forward-backward algorithm, as in (Branson et al. 2011).

On the other hand, evaluating the sum in Eq. 42 is computationally intensive. We approximate it by drawing J samples $\tilde{\theta}^t_{p1} ... \tilde{\theta}^t_{pJ}$ from the distribution $p(\tilde{\theta}_p|x, U^{t-1})$, then computing the expected entropy $p^c_{tj} = p(c|x, U^{t-1}, \tilde{\theta}^t_{pj})$ over those samples:

$$E_{\tilde{\theta}_p}[\mathrm{H}(U^{t-1}, \tilde{\theta}_p)] \approx -\sum_{j=1}^{J} p(\tilde{\theta}^t_{pj}|x, U^{t-1}) \sum_c p^c_{tj} \log p^c_{tj} \quad (43)$$

Recall that in Sect. 5.1.4 we used a similar sampling based approximation technique, where class probabilities were approximated over samples $\Theta^{t-1}_1 ... \Theta^{t-1}_K$. As in Eq. 34, we approximate $p^c_{tj} \propto p(c, U^{t-1}, \tilde{\theta}^t_{pj}|x)$ over this sample set:

$$\begin{aligned} p(c, U^{t-1}, \tilde{\theta}^t_{pj}|x) &= \int p(c, U^{t-1}, \tilde{\theta}^t_{pj}, \Theta|x) d\Theta \\ &\approx \sum_{k=1}^{K} p(c, U^{t-1}, \Theta^k, \tilde{\theta}^t_{pj}|x) = \\ p_H(U_a|c) \sum_{k=1}^{K} & p_M(c|\Theta^k, x) p_H(U_\Theta|\Theta^k) p_M(\Theta^k|x) p_H(\tilde{\theta}^t_{pj}|\theta^{t-1}_{pk}) \end{aligned} \quad (44)$$

where $p_H(\tilde{\theta}^t_{pj}|\theta^{t-1}_{pk})$ is computed using Eq. 23. The full question selection procedure is fast enough to run in a fraction of a second on a single CPU core when using 15 click questions and 312 binary questions. Fig. 11 shows a few qualitative examples of part click questions and how they are used to evolve predictions of part locations and classes.

5.2.4 Selecting Questions by Time

The expected information gain question selection method (Eq. 37) can roughly be understood as a greedy algorithm that attempts to minimize the total number of questions asked (as bits of information can be equated to binary questions).

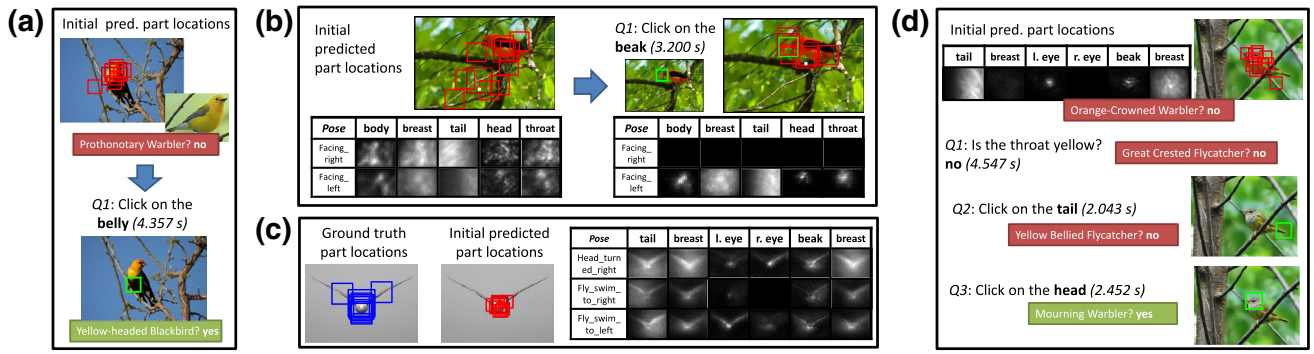


Fig. 11 Four examples of the behavior of our system. (**a**) The system estimates the bird pose incorrectly but is able to localize the head and upper body region well, and the initial class prediction captures the color of the localized parts. The user's response to the first system-selected part click question helps correct computer vision. (**b**) The bird is incorrectly detected. The system selects "*Click on the beak*" as the first question to the user. After the user's click, other part location probabilities are updated and exhibit a shift towards improved localization and pose estimation. (**c**) Certain infrequent poses (*e.g.*, flying while in frontal view) are not well captured by our detector. The initial probability distributions of part locations over the image demonstrate the uncertainty in fitting the pose models. The system tends to fail on these unfamiliar poses. **d** The system will at times select both part click and binary questions to correctly classify images

This is suboptimal as different types of questions tend to take more time to answer than others (*e.g.*, part click questions are usually faster than attribute questions). We include a simple adaptation that attempts to minimize the expected amount of human time spent. The information gain criterion $\mathrm{IG}_t(q_j)$ encodes the expected number of bits of information gained by observing the random variable u_j. We assume that there is some unknown linear relationship between bits of information and reduction in human time. The best question to ask is then the one with the largest ratio of information gain relative to the expected time to answer it:

$$q^*_{j(t+1)} = \arg\max_{q_j} \frac{\mathrm{IG}_t(q_j)}{\mathbb{E}[\mathrm{time}(u_j)]} \tag{45}$$

where $\mathbb{E}[\mathrm{time}(u_j)]$ is the expected amount of time required to answer a question q_j, which we estimate as the average response time of Mechanical Turkers.

6 Experimental Results

In this paper, we proposed several different computer vision models for multiclass classification based on shared parts and attributes (Sect. 3), human models for answering questions relating to perception of parts and attributes (Sect. 4), and a hybrid model for combining humans and computers (Sect. 5). Our experiments in this section include lesion study experiments to evaluate the utility of each component, a user study of people using a realtime implementation of our system for bird species classification, and experiments on the CUB-200-2011 (Wah et al. 2011) and Animals With Attributes (Lampert et al. 2009) datasets. Our experiments are organized as follows:

1. In Sect. 6.1, we describe implementation details for computer vision and human models
2. In Sect. 6.2, we evaluate different fully automatic computer vision algorithms on CUB-200-2011, including non-localized mutliclass methods, part-localized multiclass methods, and several different attribute-based methods. The results, implementation details, and relation to the algorithms described in this paper are summarized in Table 1.
3. In Sect. 6.3, we evaluate the effect of different human models and hybrid systems, including the relative utility of different possible strategies for picking which question to pose to humans (Fig. 12a), the relative utility of computer vision versus humans as different information sources (Fig. 12b), the relative utility of binary, multiple choice, multi-select, and part click questions (Fig. 12c), and the effect of imperfect human responses (Fig. 13a). The results, implementation details, and relation between experiments and the technical content of this paper are summarized in Table 3.
4. In Sect. 6.4, we conduct a user study of people using a realtime implementation of our system to classify bird species.
5. In Sect. 6.5, we perform additional experiments beyond bird species classification on the Animals With Attributes dataset (Lampert et al. 2009).

6.1 Implementation Details

6.1.1 CUB-200-2011 Dataset

CUB-200-2011 (Wah et al. 2011) is a dataset of 11,788 images over 200 bird species. Each image was exhaustively labeled with 15 different part locations and 312 binary attributes by Mechanical Turk workers. The dataset was divided into a training set and testset—both of 5,794 images—and a validation set of 200 images.

6.1.2 Human Models

Human user models were learned from MTurk labels on the training set. Per-class binary and multiple choice models were estimated using Eq. 19 and 21 with prior parameter $\beta = 4$. Part-click models were estimated as described in Sect. 4.2. Our 312 binary attributes were divided up both into yes/no binary questions, as well as 29 groupings (*e.g.*, *belly color* is a grouping of 15 binary attributes) that were divided into 12 multiple choice questions and 17 multi-select questions.

6.1.3 Part Detection

For part detection, we used 7×7 HOG templates for each aspect detector (mixture component), with 100 mixture components for the body, 50 mixture components for the head, and 30 mixture components for all other parts. Mixture components were learned using the procedures described in the supplementary material. All detectors were trained jointly using a structured SVM. Detection scores were converted to probabilities by optimizing Eq. 10 on our validation set.

6.1.4 Species Classification

For multiclass recognition, we extracted Fisher vector (Perronnin et al. 2010) encoded color and SIFT features. In each case, a codebook of 100 words was learned using a Gaussian mixture model. For SIFT, parameter settings and normalization schemes were performed as described in (Perronnin et al. 2010) (inducing a $2 \times 64 \times 100$-dimensional feature vector $\boldsymbol{\varphi}_{\mathbf{p}}(\theta_p; x)$ for each part p), and SIFT descriptors were extracted from patches of width 16, 24, 32, 40, and 64 pix-

Table 1 Method summary and results for automated computer vision algorithms (no human-in-the-loop) on 200 class CUB-200-2011 dataset, measured in terms of classification accuracy

Method	Class Scores $m^c(x)$	Training	Prediction	Classification Accuracy	
Non-Localized Multiclass (Sec. 3.2.1)	$\mathbf{w^c} \cdot \boldsymbol{\phi}(x)$	Eq. 3	$y = \arg\max_c m^c(x)$	28.2%	
Localized Multiclass ML Parts (Sec. 3.3.2)	$\mathbf{w^c} \cdot \boldsymbol{\Psi}(\Theta; x)$	Eq. 3 w/ $\Theta = \Theta_{GT}$	$\Theta_{ML} = \arg\max_\Theta g(\Theta; x)$ $y = \arg\max_c m^c(\Theta_{ML}; x)$	53.4%	Effect of Localization
Localized Multiclass Sample Parts (Sec. 5.1.4)	$\mathbf{w^c} \cdot \boldsymbol{\Psi}(\Theta; x)$	Eq. 3 w/ $\Theta = \Theta_{GT}$	Sample $\Theta_1 ... \Theta_K$ from $g(\Theta; x)$ $y = \arg\max_c \sum_k \exp\{\kappa m^c(\Theta_k; x) + \gamma g(\Theta_k; x)\}$	$K = 1$ 53.4% $K = 5$ 54.6% $K = 20$ 55.2% $K = 50$ 55.3%	
Localized Multiclass GT Parts (Sec. 3.3.2)	$\mathbf{w^c} \cdot \boldsymbol{\Psi}(\Theta; x)$	Eq. 3 w/ $\Theta = \Theta_{GT}$	Θ_{GT} From Oracle $y = \arg\max_c m^c(\Theta_{GT}; x)$	64.5%	
Localized Attributes (Sec. 3.4)	$\mathbf{a^c} \cdot \mathbf{m}^a(\Theta; x)$	Eq. 7 w/ $\Theta = \Theta_{GT}$	$\Theta_{ML} = \arg\max_\Theta g(\Theta; x)$ $y = \arg\max_c m^c(\Theta_{ML}; x)$	28.7%	Attribute Methods
Localized Attributes Joint Learn (Sec. 3.4)	$\mathbf{a^c} \cdot \mathbf{m}^a(\Theta; x)$	Eq. 3 w/ $\Theta = \Theta_{GT}$	$\Theta_{ML} = \arg\max_\Theta g(\Theta; x)$ $y = \arg\max_c m^c(\Theta_{ML}; x)$	43.4%	
Localized Class + Attributes (Sec. 3.4)	$\mathbf{w^c} \cdot \boldsymbol{\Psi}(\Theta; x) + \mathbf{a^c} \cdot \mathbf{m}^a(\Theta; x)$	Eq. 3 w/ $\Theta = \Theta_{GT}$	$\Theta_{ML} = \arg\max_\Theta g(\Theta; x)$ $y = \arg\max_c m^c(\Theta_{ML}; x)$	56.5%	

All methods use the same feature space, as described in Sect. 6.1. The 1st four columns provide technical details for implementation and a link to the relevant sections describing each method. The first row measures performance using an unlocalized classification model (extracting image level features); we see a significant improvement in performance from incorporating a part-localized model (28.2 → 55.3 %). The middle three rows compare different related procedures for combining part detection with multiclass recognition. We see that sampling multiple pose predictions yields a small improvement over just using the maximum likelihood prediction. The accuracy of a fully automated system (55.3 %) isn't that far behind the accuracy we would obtain if we were given ground truth part locations at test time (64.5 %), suggesting our current bottleneck is probably the performance of our part-localized classifiers/features rather than part detectors. The last three rows compare different methods for attribute-based classification. We see a big gain (28.2 → 43.4 %) from training attributes jointly (rather than independently); however, a per-class model outperforms an attribute-based one (53.4 vs. 43.4 %). A solution that combines a per-class model with an attribute model yields the best performance (56.5 %)

els. For color, we trained our codebook on 2 × 2 templates of raw pixels in Lab color space (inducing a 2 × 12 × 100-dimensional feature vector per part). In both cases, features were extracted densely from a 56 × 56 patch around each predicted part location[4], and features from all 15 parts were concatenated into one long feature vector $\boldsymbol{\Psi}(\Theta; x)$. For non-localized methods, we extracted the same set of features from the entire image, inducing a vector $\boldsymbol{\phi}(x)$. Weights for all classes were learned using a linear multiclass SVM (Eq. 3). Classifier scores were converted to probabilities by optimizing Eq. 5 on our validation set. For attribute-based methods, we used 312-dimensional soft, per-class attribute vectors $\mathbf{a^c}$ provided with the CUB-200-2011 dataset.

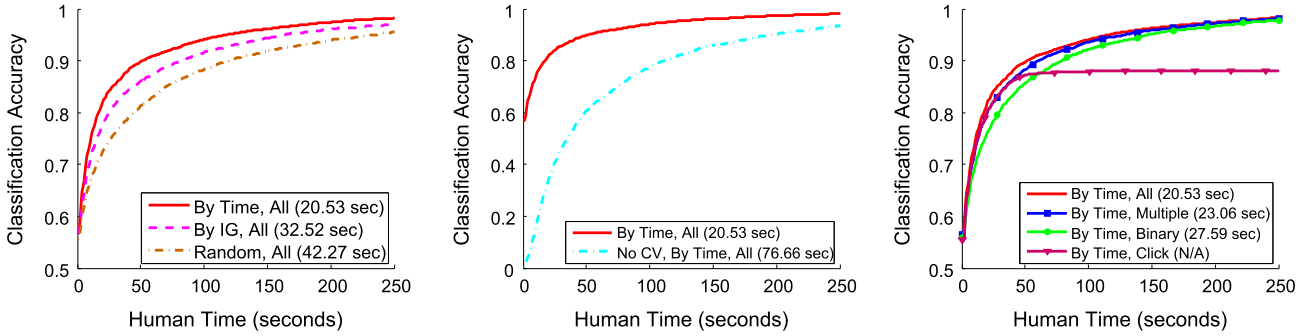
6.2 Fully Automated Computer Vision Results

In Table 1, we present classification accuracy using computer vision (with no human-in-the-loop) on the full 200 class CUB-200-2011 dataset. We compare each of the main computer vision algorithms proposed in Sect. 3 using a fixed feature space as described in Sect. 6.1, including a traditional multiclass classifier without a localization model (Sect. 3.2.1), two different variants of a multiclass classifier with a localization model based on shared parts (Sect. 3.3.2), and 3 different variants of a multiclass classifier based on shared parts and attributes (Sect. 3.4). The results, implementation details, and connection to the relevant technical sections of the paper are shown in Table 1. We summarize the results below:

6.2.1 Comparing Localization Models

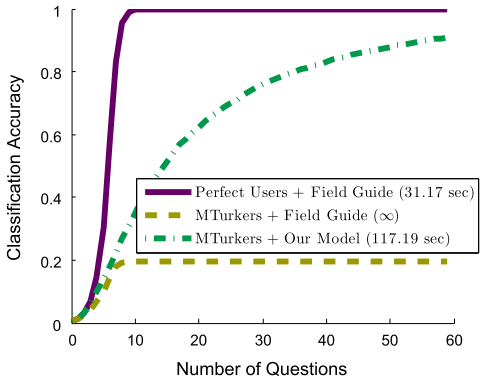
We see a significant performance increase from incorporating a localization model, from 28.2 % (traditional multiclass classifier on image-level features) to 53.4 % (a multiclass classifier on part-localized features extracted from the maximum likelihood prediction of a part-based detector). We also compare to an alternate method in which K different sets of part locations are sampled when estimating class probabilities (Sect. 5.1.4)—this results in a slight improvement in performance from 53.4 to 55.3 %, with good performance at $K = 20$ samples. If an oracle provided ground truth part locations at test time, performance could be boosted further to 64.5 %—this represents an upperbound on performance of our current features/model for multiclass classification if we had perfect part detectors.

[4] in practice, we also computed an average segmentation mask for each part-aspect and used that to weight each extracted patch, see supplementary material

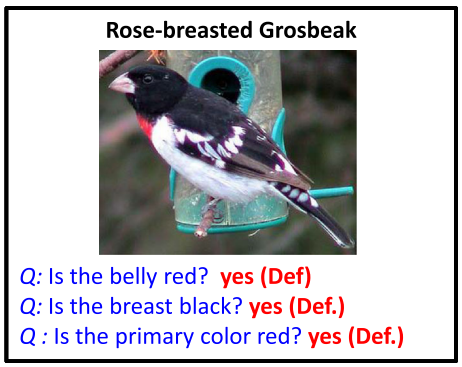


(a) Comparing Question Selection Techniques **(b)** Effect of Computer Vision **(c)** Effect of Different Types of Questions

Fig. 12 Performance on CUB-200-2011 for different crippled versions of our algorithm. Plots show how quickly classification accuracy improves as users spend more time answering questions for different methods, with the average time to correctly classify an image shown in the legend. See Table 3 for the technical details for each method. (**a**) A comparison of different question selection techniques shows that selecting by time (Eq. 45) significantly outperforms selecting by information gain, which outperforms selecting questions randomly. (**b**) Selecting by time, computer vision reduces average classification time from 76.66 to 20.53 s (*cyan* vs. *red*). (**c**) Selecting by time and using computer vision, incorporating multiple choice and multi-select questions reduces time from 27.59 to 23.06 s compared to binary questions (*green* vs. *blue*), and adding part click questions further reduces time from 23.06 to 20.53 s (*blue* vs. *red*). Note that since there are only 15 total part click questions, they aren't always sufficient to obtain perfect classification (purple curve) (color figure online)



(a) Effect of different user models



(b) Example of noisy user responses

Fig. 13 Different models of user responses. (**a**) Classification performance on CUB-200-2011 using human answers to binary attribute questions (no computer vision). Performance rises quickly (*purple curve*) if users respond deterministically according to ground truth attributes. MTurk users respond quite differently, resulting in low performance (*yellow curve*). A learned model of MTurk responses is much more robust (*green curve*). (**b**) A test image where users answer several questions incorrectly—the belly is white (*not red*), the breast is white and red (*not black*), and the primary color is white and black (*not red*)—and our model still classifies the image correctly (Color figure online)

6.2.2 Comparing Attribute-Based Methods

We implementated a few different part-localized attribute-based methods. These three attribute methods include (1) a method in which attribute classifiers are trained independently and then combined probabilistically [this is the traditional approach, as in (Lampert et al. 2009], (2) a method in which attribute weights are learned jointly to optimize multi-class classification accuracy, and (3) a method that combines both per-class and per-attribute weights, both of which are learned jointly. The technical details of each method are summarized in the last three rows of Table 1. Our results show that learning attributes jointly (instead of independently) significantly improves classification accuracy, from 28.7 to 43.4 %; however, both attribute-based methods do not perform as well as a per-class model, which achieves an accuracy of 53.4 %. A possible explanation is that a low-dimensional vector of class-attribute memberships is not sufficiently discriminative to distinguish bird species. A model that combines per-class weights (which may better capture fine-grained differences between classes) and per-attribute weights (which may improve generalization when the number of training examples is small) outperforms all methods, achieving an accuracy of 56.5 %.

Table 2 Comparison to related work (computer vision) on 200 class CUB-200-2011 dataset, measured in terms of classification accuracy

Method	Accuracy	
SIFT+color+SVM [69]	10.3%	Earlier methods
Pose Pooling Kernels [76]	28.2%	
Part 1-vs.-1 Features (POOF) [2]	56.8%	Recent methods
Deformable Part Descriptors [77]	51.0%	
Symbiotic Segmentation & Parts [8]	59.4%	
FGVC by Alignments [22]	62.7%	
Localized Multiclass+Attributes	56.5%	Ours

A number of papers (Berg and Belhumeur 2013; Zhang et al. 2013; Chai et al. 2013; Gavves et al. 2013) have recently come out in CVPR and ICCV 2013 that significantly outperform earlier methods on CUB-200-2011. Like our paper, these papers combine newer features with an improved localized model

6.2.3 Comparison to Other Papers

Table 2 shows a comparison of our computer vision performance to other papers. A number of papers (Berg and Belhumeur 2013; Zhang et al. 2013; Chai et al. 2013; Gavves et al. 2013) have recently come out in CVPR and ICCV 2013 that obtain classification accuracies of $51--62\,\%$ on CUB-200-2011, a significant improvement over the results of earlier published work [$10.3--28.2\,\%$ (Wah et al. 2011; Zhang et al. 2012)]. These papers employ similar algorithms to our paper: newer features with an improved part-based localization model. Our performance is in the same realm as these newer papers.

6.3 Simulated Human-in-the-Loop Experiments from MTurk Responses

Although we have put effort into developing high performing computer vision algorithms, the point of this paper is to introduce algorithms for human-in-the-loop systems. The experiments in the remaining sections focus on interactive algorithms for improving classification accuracy while minimizing human time.

To compare different versions of our algorithms, we exhaustively collected answers to all image-question pairs using Mechanical Turk using the GUIs shown in Figs. 3, 4. We used the resulting answers and response times to simulate human-in-the-loop classification sessions using the following procedure:

1. Predict the class with highest probability $p(c|x, U)$ according to Eq. 34. If the predicted class is the true class, assume the simulated user will stop the interface (*e.g.*, by verifying the correctness of the predicted class after being shown a small gallery of images).
2. Select a question to pose to the user and lookup the answer and response time from the corresponding MTurk experiment.
3. Repeat steps 1–2 until the user stops the interface

We measure the average total human time spent per test image, excluding the time it takes to verify correctness of the species (which we did not measure). Note that we have assumed that people are perfect verifiers, *e.g.*, they will stop the system if and only if they have been presented with the correct class. We explore the legitimacy of this assumption on real-life user studies in Sect. 6.4. We performed simulated experiments for different ways of computing $p(c|U, x)$ for different lesioned versions of our algorithms and different criteria for selecting the next question. The technical details for each experiment are shown in Table 3. We summarize the results in the subsections below:

6.3.1 Question Selection by Time Reduces Human Effort

In Fig. 12a, we compare three different question selection techniques: Random (choosing a random question among multiple choice, multi-select, and click questions but excluding binary questions), expected information gain (Eq. 37), and time (Eq. 45). For fairness, we excluded binary questions from random selection; they are almost never useful because they are redundant with multiple choice questions while providing a subset of the information, and there are far more binary questions than multiple choice questions (such that selecting a question uniformly at random would favor picking binary ones). We see that the information gain criterion reduces average time from 42.27 to 32.52 s, whereas selecting by time results in a reduction to 20.53 s. Note that we have reduced our classification time from 58.4 to 20.53 s compared to an earlier version of our algorithms (Wah et al. 2011); this is primarily a result of improved computer vision algorithms and incorporation of multi-select questions.

6.3.2 Computer Vision Reduces Manual Labor

The main benefit of computer vision is that it reduces the amount of human time needed to identify the true species. In Fig. 12b, we see that computer vision reduces the average time from 76.66 to 20.53 s when choosing questions by time.

6.3.3 Multiple Choice and Multi-Select Questions are Useful

In Fig. 12c, we compare results when certain types of questions are removed. We see that using multiple choice and multi-select questions reduces average time from 27.59 to 23.06 s compared to using binary questions.

Table 3 Method summary and results for recognition with a human-in-the-loop on 200 class CUB-200-2011 dataset, measured in terms of amount of human time to identify the true class

Method	Binary	Multiple Choice	Click	Computer Vision	Question Select	Users	Train User Responses	Human Time	
By Time, All	✓	✓	✓	✓	Time (Eq. 45)	MTurk	Eq. 19,21,23	20.53 sec	Fig. 12(a)
By IG, All	✓	✓	✓	✓	IG (Eq. 37)	MTurk	Eq. 19,21,23	32.25 sec	
Random, All	✓	✓	✓	✓	Random	MTurk	Eq. 19,21,23	42.27 sec	
No CV, By Time, All	✓	✓	✓	✗	Time (Eq. 45)	MTurk	Eq. 19,21,23	76.66 sec	Fig. 12(b)
By Time, Binary	✓	✗	✗	✓	Time (Eq. 45)	MTurk	Eq. 19	27.59 sec	Fig. 12(c)
By Time, Multiple	✗	✓	✗	✓	Time (Eq. 45)	MTurk	Eq. 21	23.06 sec	
By Time, Click	✗	✗	✓	✓	Time (Eq. 45)	MTurk	Eq. 23	N/A	
Perfect Users + Field Guide	✓	✗	✗	✗	Time (Eq. 45)	Oracle $\tilde{a}_i = a_i^c$	$\hat{p}_i^c = a_i^c$	31.17 sec	Fig. 13(a)
MTurkers + Field Guide	✓	✗	✗	✗	Time (Eq. 45)	MTurk	$\hat{p}_i^c = a_i^c$	∞	
MTurkers + Our Model	✓	✗	✗	✗	Time (Eq. 45)	MTurk	Eq. 19	117.2 sec	

For all methods, class probabilities $p(c|U, x)$ were computed using Eq. 34, with different components of the model removed as indicated by rows 2–5. All methods that use computer vision use the *Localized Multiclass, Sample Parts* method with $K = 20$ (3rd row in Table 1) to estimate class probabilities $p_M(c|\Theta^k, x)$ and part probabilities $p_M(\Theta^k|x)$. Methods that incorporate multiple choice or binary questions use attribute response probabilities $p_H(U_a|c) = \prod_i p_H(\tilde{a}_i|c)$ (see Sects. 4.1.1–4.1.2), and methods that incorporate click questions use click response probabilities $p_H(U_\Theta|\Theta^k) = \prod_p p_H(\tilde{\theta}_p|\theta_p)$ (see Sect. 4.2). The 8th column shows which equations were used to train the user (human) model. The 6th column shows the criterion used to choose which question to pose to the user

6.3.4 Click Questions are Asked Early, if Ever

In Fig. 12c, we see that adding click questions in addition to multiple choice questions reduces average classification time from 23.06 to 20.53 s. We note that multiple choice questions are overwhelmingly favored over binary questions [see Fig. 14]. For the first question, it chooses a click question versus a multiple choice question with roughly equal probability; however, it almost never chooses a click question again until the most useful multiple choice questions have been exhausted. This most likely occurs because the localization mistakes that are most critical to classification error are typically corrected by one part click question.

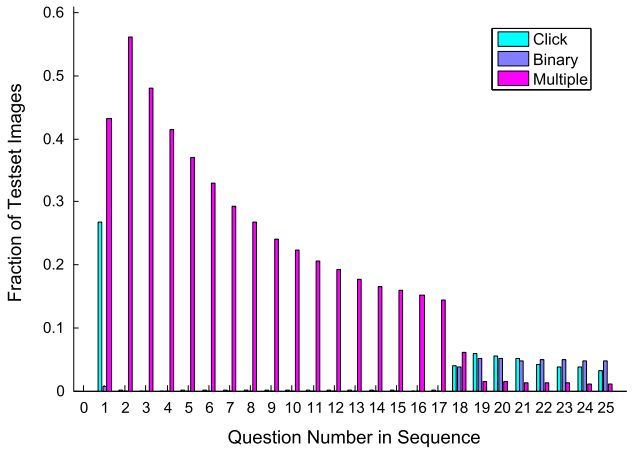


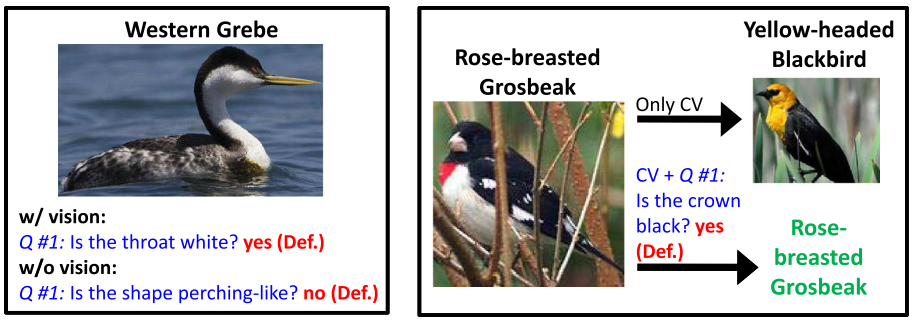
Fig. 14 Analysis of when different types of questions were usually selected. Click questions were usually chosen as the first question (if at all), after which multiple choice/multi-select questions were heavily favored. Notice that the majority of queries end within a few questions

6.3.5 User Responses are Stochastic

In Fig. 13a, we explore the effect of different user models, without any computer vision in the loop [see the last three rows of Table 3 for technical details]. When users are assumed to respond deterministically in accordance with groundtruth class-attributes, performance rises quickly to 100 % within 8 binary questions (roughly $\log_2(200)$). However, this assumption is not realistic; when testing with responses from MTurk, performance saturates at around 20 %. Subjective answers are unavoidable (*e.g.*, perception of the color brown versus the color buff), and the probability of the correct class drops to zero after any inconsistent response. Although performance is 40 times better than random chance, it renders the system useless. This demonstrates a challenge for existing field guide websites. When our learned model of user responses [(see Sect. 4.1] is incorporated, performance keeps improving as more binary questions are answered due to the ability to tolerate a reasonable degree of error in user responses (see Fig. 13b, c). Nevertheless, stochastic user responses significantly increase the number of questions required to achieve a given accuracy level.

6.3.6 Different Questions are Asked with and Without Computer Vision

In general, the information gain criterion favors questions that (1) can be answered reliably, and (2) split the set of pos-



(a) Different questions w/ vision **(b)** User helps correct mistakes of vision

Fig. 15 Qualitative examples. (**a**) An image that is only classified correctly when computer vision is incorporated. Additionally, the computer vision based method selects the question *is the throat white*, a different and more relevant question than when vision is not used. (**b**) the user response to *is the crown black* helps correct computer vision when its initial prediction is wrong

sible classes roughly in half. Binary attributes like *perching-like shape*, which divide the classes fairly evenly, and *yellow underparts*, which tends to be answered reliably, are commonly chosen. When computer vision is incorporated, the likelihood of classes changes and different questions are selected. In the left image of Fig. 15, we see an example where a different question is asked with and without computer vision, which allows the system to find the correct class using one question. The left image in Fig. 15 shows an example of an image classified correctly using computer vision, which is not classified correctly without computer vision, even after asking 60 questions.

6.4 User Study

The results in previous sections were simulated using question responses of Mechanical Turk users. Doing this allowed us to systematically test different variations of our algorithms without re-running human-in-the-loop experiments; however, certain aspects of a real life interface were lost in simulation. We ran a study of users using a full fledged web-based version of our tool to interactively identify birds. The web-based tool, shown in Fig. 16a, communicates with a server that runs computer vision algorithms. A single desktop computer with a 2.9GHz quadcore CPU was able to handle at least eight simultaneous users (we didn't try more) while serving all requests in about 1 s or less. The user study was conducted when our computer vision algorithms were only 30 % accurate (we since tweaked scale parameters of SIFT features to improve performance to 55 %). A screen capture of the user study interface is included as a supplementary video.

In this study, 27 human subjects were each asked to use our tool to identify 10 bird images that were randomly selected from the CUB-200-2011 test set. Of these 27 subjects, 20 had no experience in birding or using our interface. Among these 20 inexperienced users, the average time to identify a bird was 73.7 s, with an average classification accuracy of 54 %, and average taxonomic loss of .99. The taxonomic loss is defined as the distance to the closest common ancestor of a predicted species and ground truth species according to scientific classification (species, genus, family, order, class). We discuss additional details and analysis of the user study in the sub-sections below:

6.4.1 The Verification Problem

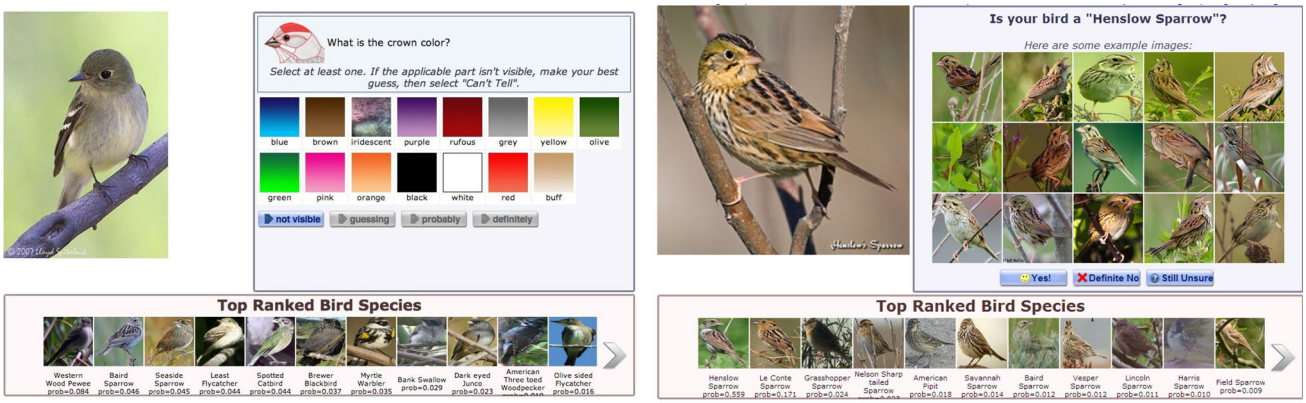
There is an additional challenge of verifying correctness of a predicted species that was not modeled in our synthesized experiments. We handled this by adding a verification capability to our GUI, where a user could click on a thumbnail of a top-ranked species to examine a set of exemplars (see Fig. 16b). The user can then make a decision of whether or not to stop the interface by choosing that species. The verification process introduces new sources of time and error (recall that for simulation in Sect. 6.3, we assumed users were perfect verifiers).

6.4.2 The Tradeoff Between Time and Accuracy

There is an inherent tradeoff between classification accuracy and time; greater accuracy can be achieved by spending more time answering questions and exploring the verification interface. A user seeking to identify her own uploaded picture of a bird will tradeoff these two things according to her own preferences. By contrast, our users were not interested in recognizing birds. To put each user on equal footing, we primed them with a loss function

$$\text{loss} = \text{taxonomic loss} + \frac{\text{time in seconds}}{45} \tag{46}$$

Users were instructed to try to minimize this loss. In Fig. 17a, the diagonal blue line depicts an equal level of loss due to time and taxonomic loss. Each point depicts a different user



(a) Web Interface **(b)** Verification Interface

Fig. 16 Web-based interface. Screen captures of the web-interface used to conduct our user study. (**a**) The web-interface shows a query image (*left*), a question (*right*), and the 10 top-ranked bird species (*bottom*). The user can either choose to answer the question or click on the bottom thumbnail of the bird species. (**b**) When the user clicks on one of the top-ranked bird species, a verification interface is opened. The user can examine additional exemplars and decide whether or not it is the same species as in the query image. See the supplementary material for a video of a user using the web-interface.

(where his/her loss was averaged over 10 images). Points closer to the origin indicate users with lower combined loss.

6.4.3 Improvement Over Existing Online Field Guides

A subset of five users were asked to identify additional images using the online field guide website whatbird.com. All users were able to identify birds much more quickly and accurately using our interface (on average, a time reduction from 219 to 73.7 s, and a taxonomic loss reduction from 2.12 to .99) as seen in Fig. 17a. Although the sample size was low, users agreed that our interface clearly offered significant improvements due to not assuming question answers are deterministic and incorporating computer vision.

6.4.4 Familiarity with Our UI Affects Classification Time

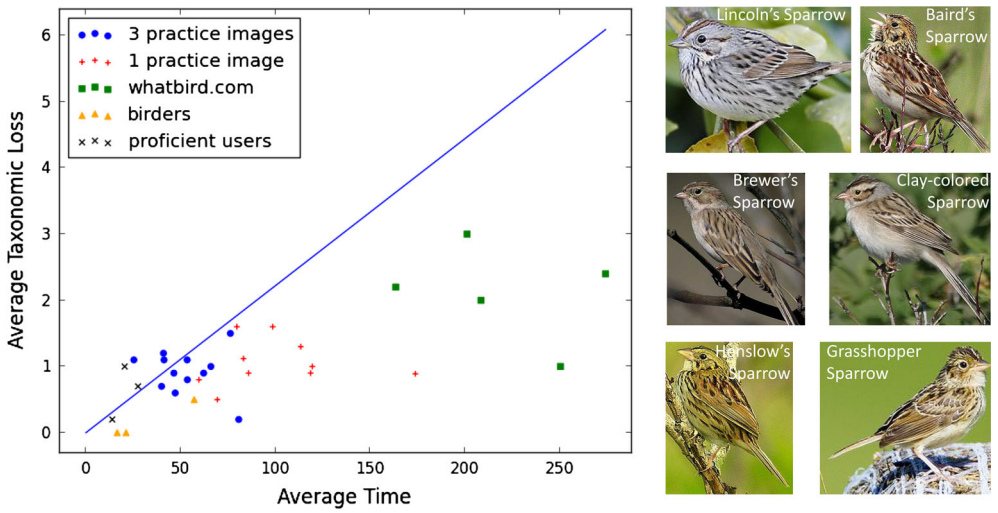
In Fig. 17a, we plot 27 different users in terms of their average time and average taxonomic loss. 20 such users were young computer science students with no background in birding or experience using our interface. These 20 users were divided into two groups, the first of which was given 1 training image to become familiar with our interface before starting the experiment, and 2nd of which was given three training images. The group that was given more training images was able to identify birds much faster (on average, 52.7 vs. 99.6 s) with similar level of classification error. Gaining greater familiarity with the interface reduces classification time because users spend less time reading instructions for each question, and have more familiarity with the relative tradeoff between answering more questions or browsing through the verification interface. Similarly supporting this claim, three users who were proficient with the interface but not very familiar with birding (the three student authors of this paper) were able to identify birds in 20.6 s on average while also being slightly more accurate.

6.4.5 Birding Experience and Sources of Classification Error

We additionally performed our study on three expert birders, two of which are considered to be among the top birders in the world. These birders were able to identify birds both quickly and accurately, with an average classification accuracy of 93 % in 31.9 s. By contrast, the average accuracy of non-birders was 54 %. The primary reason for this discrepency is that non-birders have no prior knowledge of the space of birds or the relatedness of different species. Thus when presented with an incorrect but similar bird species (*e.g.*, consider the different sparrow species shown in Fig. 17b), the users were likely to choose the wrong one. An additional problem is that when bird species appear very similar, some bird species are not separable in attribute space with high probability (since attribute responses can be noisy/subjective). In this case, the best the interface can do is to communicate a set of candidate species that are consistent with both attribute responses and computer vision.

According the the Cornell Ornithology Website[5], the four keys to bird species recognition are (1) size and shape, (2) color and pattern, (3) behavior, and (4) habitat. Bird species classification is a difficult problem and is not always possible using a single image. One potential advantage of the visual 20 questions formulation is that other contextual sources of information such as behavior and habitat can easily be incor-

[5] http://www.allaboutbirds.org/NetCommunity/page.aspx?pid=1053



(a) User Study Results **(b)** Sparrow Species

Fig. 17 User study results. 27 different users used our interface to identify 10 bird images. (**a**) Each point plots the average classification time (x-axis) versus average taxonomic loss (y-axis) for a particular user. See Sect. 6.4 for a definition of taxonomic loss. Users were instructed to optimize a combined loss that trades off time and taxonomic loss (Eq. 46), with the blue line depicting equal loss due to the two considerations. Most users (*blue dots* and *red plus symbols*) had no prior experience birding or using our interface. Users given three training images before starting the experiment (*blue dots*) were significantly faster than users given one training image (*red plus symbols*). Users who were non-birders but had prior experience using our interface (*black* X's) were even faster. Users who were expert birders (*orange triangles*) were both fast and accurate. All users of our interface were significantly faster and more accurate using our interface than users using whatbird.com (*green squares*). (**b**) Images of different sparrow species appear similar to non-bird experts, such that users are likely to stop the interface early or choose the wrong one. This is one of the main reasons why users don't get 100 % classification accuracy. As a reference point, the average pairwise taxonomic loss between species in this cluster of sparrows is 1.73 (*i.e.*, all come from the family Emberizidae, while most do not share the same Genus) (color figure online)

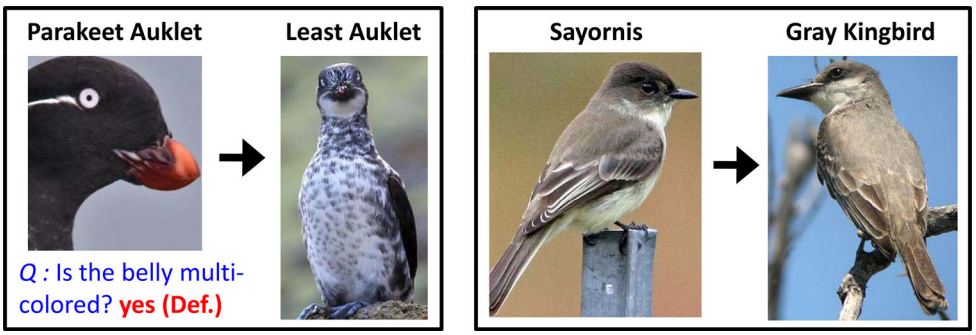
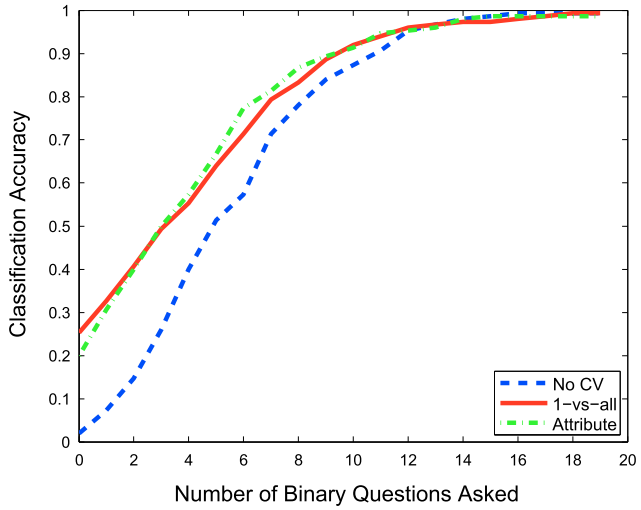


Fig. 18 Images that are misclassified by our system. In each of the two panels, the left image is a query image that a user classified using our system, and the right image is an exemplar of an incorrect species prediction. *Left* The Parakeet Auklet image is misclassified due to a cropped image, which causes an incorrect answer to the belly pattern question (the Parakeet Auklet has a plain, white belly). *Right* The Sayornis and Gray Kingbird are commonly confused due to visual similarity

porated as additional questions. Figure 18 illustrates some example failures.

6.5 Animals with Attributes Dataset

Animals with attributes (AwA) is a dataset of 50 animal classes such as zebras, pandas, and dolphins. Each class is associated with soft labels for 85 binary attributes based on posing class-level attribute questions to multiple people, effectively encoding a distribution $p(\tilde{a}_i|c)$. We simulate test performance by randomly selecting a question response based on $p(\tilde{a}_i|c)$. The dataset also includes class-attribute labels, which were obtained by thresholding the soft labels. While the dataset is not exactly aligned with our goal of recognition of finer-grained categories, it is the most established dataset with the types of annotations required for our application outside of CUB-200-2011. The dataset is difficult due to large intraclass variation and unaligned images. We train unlocalized multiclass (Sect. 3.2.1) and attribute-based (Sect. 3.2.2) computer vision algorithms using precomputed features packaged with the dataset. The results of our experiments using simulated user responses are shown in Fig. 19.



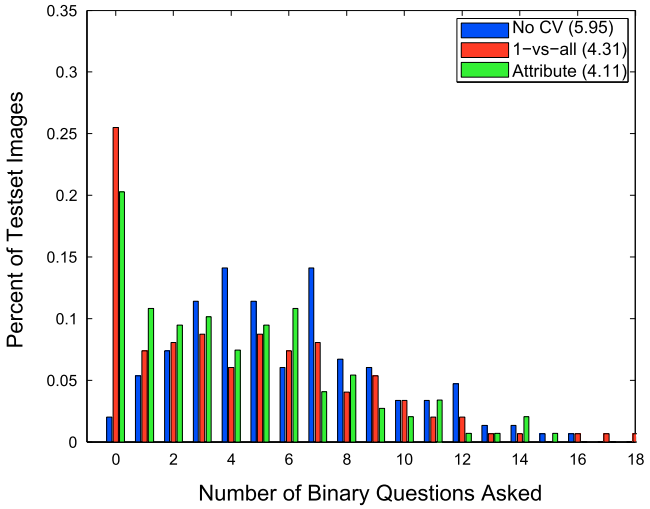


Fig. 19 Performance on animals with attributes (probabilistic attributes). *Left Plot* Classification performance, simulating user responses using soft class-attributes [see (Lampert et al. 2009]. *Right Plot* The required number of questions needed to identify the true class drops from 5.94 to 4.11 on average when incorporating computer vision

7 Conclusion

Object recognition remains a challenging problem for computer vision. Furthermore, recognizing amongst fine-grained categories is difficult even for humans. While neither humans nor computers excel at the task, their abilities and failings are complementary. Humans can detect objects and classify them into broad categories; they can also locate object parts and measure attributes, such as color and shape. Machines can remember and handle complex taxonomies, as well as the association between categories and attributes, and can accurately compute probabilities of classifications based on the value of those attributes.

We propose a hybrid human–machine visual system that combines the strengths of both: the human can see better, the machine can better classify, ask the right questions and integrate information. The design is simple: it is based on iterating a sequence of four steps: (a) the machine's visual system is used to detect the object and its parts, and to measure its attributes, based on the available information (image and human answers); (b) the machine updates its estimate of the probability of each category; (c) the machine selects the most informative question that the human should address; (d) the human answers the question. This design is modular and may be used in conjunction with a large variety of computer vision algorithms.

In order to test our design we implemented a field guide for identifying birds. The field guide was trained using data collected from paid annotators who answered both click and attribute questions in response to test images of specimens belonging to each one of 200 species of birds. We carried out two experiments: (a) using a set of annotator responses which had not been used in training, to simulate the responses of a putative user, (b) with real users who were challenged to identify bird species in the least amount of time using a real-time version of our system.

First of all, our experiments show that our subordinate categorization computer vision system is about 56 % correct when operating in isolation, without the help of a user. This is state-of-the-art performance that was achieved after 3 years of research in algorithms for fine-grained recognition; this performance is still lower than what we would like for a useful application.

Second, we find that existing field guides that use attribute queries to deterministically index into a database of bird species are mostly unusable by non-experts. Users' responses to attribute questions vary a lot due to subjective differences and often do not agree with expert-defined attributes. A probabilistic model of human attribute responses leads to significantly better classification performance in comparison to deterministic field guides generated by experts.

Third, we find that a hybrid system that combines machine vision with user input drives up performance. The combination of machine and human is not purely a combination of machine and human sensors. Rather, the machine dynamically selects the most informative questions to be asked of the human observer in order to achieve a reliable answer in the shortest amount of time.

Fourth, a real-time implementation of our bird guide is a practical and enjoyable tool for humans to achieve bird classification. The average classification error is small and classification is done quickly. In sum, our on-line bird guide is already a useful tool.

The most obvious next step for our research is to validate our ideas in other domains, besides birds. Obtaining a set of reasonable attributes and questions for the bird dataset

was relatively easy, as we relied on existing field guides. The question is open on how to infer attributes for domains where field guides are not available.

References

Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S.& Zhang, L. (2008). Searching the world's herbaria. In *ECCV*.

Berg, T.& Belhumeur, P.N. (2013). Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*.

Biederman, I., Subramaniam, S., Bar, M., Kalocsai, P., & Fiser, J. (1999). Subordinate-level object classification reexamined. *Psychological Research*, *63*(2–3), 131–153.

Bourdev, L.& Malik, J. (2009). Poselets: Body part detectors trained using 3d annotations. In *ICCV*.

Branson, S., Perona, P.& Belongie, S. (2011). Strong supervision from weak annotation. In *ICCV*.

Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P.& Belongie, S. (2010). Visual recognition with humans in the loop. In *ECCV*.

Chai, Y., Lempitsky, V.& Zisserman, A. (2011). Bicos: A bi-level co-segmentation method. In *ICCV*.

Chai, Y., Lempitsky, V.& Zisserman, A. (2013). Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*.

Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L.& Zisserman, A. (2012). Tricos. In *ECCV*.

Cox, I.J., Miller, M.L., Minka, T.P., Papathomas, T.V.& Yianilos, P.N. (2000). The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. Image processing.

Donahue, J.& Grauman, K. (2011). Annotator rationales for visual recognition. In *ICCV*.

Douze, M., Ramisa, A.& Schmid, C. (2011). Combining attributes and fisher vectors for efficient image retrieval. In *CVPR*.

Duan, K., Parikh, D., Crandall, D.& Grauman, K. (2012). Discovering localized attributes for fine-grained recognition. In *CVPR*.

Fang, Y.& Geman, D. (2005). Experiments in mental face retrieval. In *AVBPA*.

Farhadi, A., Endres, I.& Hoiem, D. (2010). Attribute-centric recognition for generalization. In *CVPR*.

Farhadi, A., Endres, I., Hoiem, D.& Forsyth, D. (2009). Describing objects by attributes. In *CVPR*.

Farrell, R., Oza, O., Zhang, N., Morariu, V., Darrell, T.& Davis, L. (2011). Birdlets. In *ICCV*.

Felzenszwalb, P.& Huttenlocher, D. (2002). Efficient matching of pictorial structures. In *CVPR*.

Felzenszwalb, P., McAllester, D.& Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *CVPR*.

Ferecatu, M.& Geman, D. (2007). Interactive search by mental matching. In *ICCV* .

Ferecatu, M.& Geman, D. (2009). A statistical framework for image category search from a mental picture. In *PAMI*.

Gavves, E., Fernando, B., Snoek, C., Smeulders, A.& Tuytelaars, T. (2013). Fine-grained categorization by alignments. In *ICCV*.

Geman, D.& Jedynak, B. (1993). *Shape recognition and twenty questions*. Belmont: Wadsworth.

Geman, D.& Jedynak, B. (1996). An active testing model for tracking roads in satellite images. In *PAMI*.

Jedynak, B., Frazier, P. I., & Sznitman, R. (2012). Twenty questions with noise: Bayes optimal policies for entropy loss. *Journal of Applied Probability*, *49*(1), 114–136.

Khosla, A., Jayadevaprakash, N., Yao, B.& Li, F.F. (2011). *Novel dataset for fgvc: Stanford dogs*. San Diego: CVPR Workshop on FGVC.

Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D., Kress, W., Lopez, I.& Soares, J. (2012). Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*.

Kumar, N., Belhumeur, P.& Nayar, S. (2008). Facetracer: A search engine for large collections of images with faces. In *ECCV*.

Kumar, N., Berg, A.C., Belhumeur, P.N.& Nayar, S.K. (2009). Attribute and simile classifiers for face verification. In *ICCV*.

Lampert, C., Nickisch, H.& Harmeling, S. (2009). Learning to detect unseen object classes. In *CVPR*.

Larios, N., Soran, B., Shapiro, L.G., Martinez-Munoz, G., Lin, J.& Dietterich, T.G. (2010). Haar random forest features and svm spatial matching kernel for stonefly species identification. In *ICPR*.

Lazebnik, S., Schmid, C.& Ponce, J. (2005). A maximum entropy framework for part-based texture and object recognition. In *ICCV*.

Levin, A., Lischinski, D.& Weiss, Y. (2007). A closed-form solution to natural image matting. In *PAMI*.

Liu, J., Kanazawa, A., Jacobs, D.& Belhumeur, P. (2012). Dog breed classification using part localization. In *ECCV*.

Lu, Y., Hu, C., Zhu, X., Zhang, H.& Yang, Q. (2000). A unified framework for semantics and feature based relevance feedback in image retrieval systems. In *ACM Multimedia*.

Maji, S. (2012). Discovering a lexicon of parts and attributes. In *ECCV Parts and Attributes*.

Maji, S.& Shakhnarovich, G. (2012). Part annotations via pairwise correspondence. In *Conference on Artificial Intelligence Workshop*.

Martınez-Munoz et al. (2009). Dictionary-free categorization of very similar objects. In *CVPR*.

Mervis, C. B., & Crisafi, M. A. (1982). Order of acquisition of subordinate-, basic-, and superordinate-level categories. *Child Development*, *53*(1), 256–266.

Nilsback, M.& Zisserman, A. (2008). Automated flower classification. In *ICVGIP*.

Nilsback, M.E.& Zisserman, A. (2006). A visual vocabulary for flower classification. In *CVPR*.

Ott, P.& Everingham, M. (2011). Shared parts for deformable part-based models. In *CVPR*.

Parikh, D.& Grauman, K. (2011). Interactively building a vocabulary of attributes. In *CVPR*.

Parikh, D.& Grauman, K. (2011). Relative attributes. In *ICCV*.

Parikh, D.& Grauman, K. (2013). Implied feedback: Learning nuances of user behavior in image search. In *ICCV*.

Parikh, D.& Zitnick, C.L. (2011a). Finding the weakest link in person detectors. In *CVPR* .

Parikh, D.& Zitnick, C.L. (2011b). Human-debugging of machines. In *NIPS Wisdom of Crowds*.

Parkash, A.& Parikh, D. (2012). Attributes for classifier feedback. In *ECCV*.

Parkhi, O., Vedaldi, A., Zisserman, A.& Jawahar, C. (2012). Cats and dogs. In *CVPR*.

Parkhi, O.M., Vedaldi, A., Jawahar, C.& Zisserman, A. (2011). The truth about cats and dogs. In *ICCV*.

Perronnin, F., Sánchez, J.& Mensink, T. (2010). Improving the fisher kernel. In *ECCV*.

Platt, J.C. (1999). Probabilistic outputs for svms. In *ALMC*.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Burlington: Morgan Kaufmann.

Rasiwasia, N., Moreno, P.J.& Vasconcelos, N. (2007). Bridging the gap: Query by semantic example. In *Multimedia*.

Rosch, E. (1999). Principles of categorization. In *Concepts: Core readings*.

Rosch, E., Mervis, C.B.& Gray, W.D., Johnson, D.M., Boyes-Braem, P. (1976). Basic objects in natural categories. In *Cognitive Psychology*.

Rother, C., Kolmogorov, V.& Blake, A. (2004). Grabcut: Interactive foreground extraction. In *TOG*.
Settles, B. (2008). *Curious machines: Active learning with structured instances*.
Stark, M., Krause, J., Pepik, B., Meger, D., Little, J.J., Schiele, B.& Koller, D. (2012). Fine-grained categorization for 3d scene understanding. In *BMVC*.
Sznitman, R., Basu, A., Richa, R., Handa, J., Gehlbach, P., Taylor, R.H., Jedynak, B.& Hager, G.D. (2011). Unified detection and tracking in retinal microsurgery. In *MICCAI*.
Sznitman, R.& Jedynak, B. (2010). Active testing for face detection and localization. In *PAMI*.
Tsiligkaridis, T., Sadler, B.& Hero, A. (2013). A collaborative 20 questions model for target search with human-machine interaction. In *ICASSP*.
Tsochantaridis, I., Joachims, T., Hofmann, T.& Altun, Y. (2006). Large margin methods for structured and interdependent output variables. In *JMLR*.
Vijayanarasimhan, S.& Grauman, K. (2009). What's It Going to Cost You? In *CVPR*.
Vijayanarasimhan, S.& Grauman, K. (2011). Large-scale live active learning. In *CVPR*.
Vondrick, C.& Ramanan, D. (2011). Video Annotation and Tracking with Active Learning. In *NIPS*.
Vondrick, C., Ramanan, D.& Patterson, D. (2010). Efficiently scaling up video annotation. In *ECCV*.
Wah, C., Branson, S., Perona, P.& Belongie, S. (2011). Multiclass recognition and part localization with humans in the loop. In *ICCV*.
Wah, C., Branson, S., Welinder, P., Perona, P.& Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, Pasadena: Caltech.
Wang, G.& Forsyth, D. (2009). Joint learning of visual attributes, object classes. In ICCV.
Wang, J., Markert, K.& Everingham, M. (2009). Learning models for object recognition from natural language descriptions. In *BMVC*.
Wu, W.& Yang, J. (2006). SmartLabel: an object labeling tool. In *Multimedia*.
Yang, Y.& Ramanan, D. (2011). Articulated pose estimation using mixtures of parts. In *CVPR*.
Yao, B., Bradski, G., Fei-Fei, L.: A codebook and annotation-free approach for fgvc. In: CVPR (2012)
Yao, B., Khosla, A.& Fei-Fei, L. (2011). Combining randomization and discrimination for fgvc. In *CVPR*.
Zhang, N., Farrell, R.& Darrell, T. (2012). Pose pooling kernels for sub-category recognition. In *CVPR*.
Zhang, N., Farrell, R., Iandola, F.& Darrell, T. (2013). Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*.
Zhou, X.& Huang, T. (2003). Relevance feedback in image retrieval. In *Multimedia*.