Visual Recognition with Humans in the Loop

Steve Branson[1], Catherine Wah[1], Florian Schroff[1], Boris Babenko[1],
Peter Welinder[2], Pietro Perona[2], and Serge Belongie[1]

[1] University of California, San Diego
{sbranson,cwah,gschroff,bbabenko,sjb}@cs.ucsd.edu
[2] California Institute of Technology
{welinder,perona}@caltech.edu

Abstract. We present an interactive, hybrid human-computer method for object classification. The method applies to classes of objects that are recognizable by people with appropriate expertise (*e.g.*, animal species or airplane model), but not (in general) by people without such expertise. It can be seen as a visual version of the *20 questions game*, where questions based on simple visual attributes are posed interactively. The goal is to identify the true class while minimizing the number of questions asked, using the visual content of the image. We introduce a general framework for incorporating almost any off-the-shelf multi-class object recognition algorithm into the visual 20 questions game, and provide methodologies to account for imperfect user responses and unreliable computer vision algorithms. We evaluate our methods on *Birds-200*, a difficult dataset of 200 tightly-related bird species, and on the *Animals With Attributes* dataset. Our results demonstrate that incorporating user input drives up recognition accuracy to levels that are good enough for practical applications, while at the same time, computer vision reduces the amount of human interaction required.

1 Introduction

Multi-class object recognition has undergone rapid change and progress over the last decade. These advances have largely focused on types of object categories that are easy for humans to recognize, such as motorbikes, chairs, horses, bottles, *etc.* Finer-grained categories, such as specific types of motorbikes, chairs, or horses are more difficult for humans and have received comparatively little attention. One could argue that object recognition as a field is simply not mature enough to tackle these types of finer-grained categories. Performance on basic-level categories is still lower than what people would consider acceptable for practical applications (state-of-the-art accuracy on Caltech-256[1] is $\approx 45\%$, and $\approx 28\%$ in the 2009 VOC detection challenge [2]. Moreover, the number of object categories in most object recognition datasets is still fairly low, and increasing the number of categories further is usually detrimental to performance [1].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 438–451, 2010.

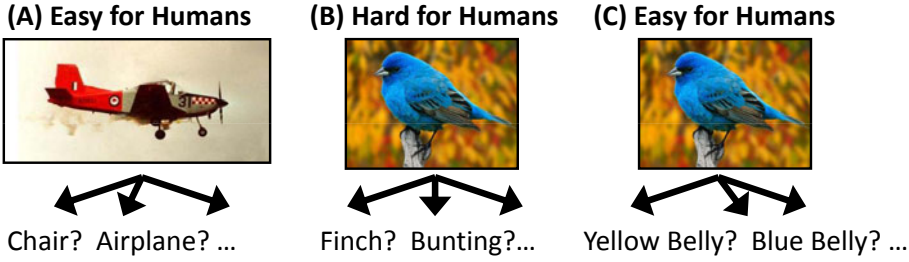

Fig. 1. Examples of classification problems that are easy or hard for humans. While basic-level category recognition (left) and recognition of low-level visual attributes (right) are easy for humans, most people struggle with finer-grained categories (middle). By defining categories in terms of low-level visual properties, hard classification problems can be turned into a sequence of easy ones.

On the other hand, recognition of finer-grained subordinate categories is an important problem to study – it can help people recognize types of objects they don't yet know how to identify. We believe a hybrid human-computer recognition method is a practical intermediate solution toward applying contemporary computer vision algorithms to these types of problems. Rather than trying to solve object recognition entirely, we take on the objective of minimizing the amount of human labor required. As research in object recognition progresses, tasks will become increasingly automated, until eventually we will no longer need humans in the loop. This approach differs from some of the prevailing ways in which people approach research in computer vision, where researchers begin with simpler and less realistic datasets and progressively make them more difficult and realistic as computer vision improves (*e.g.*, Caltech-4 → Caltech-101 → Caltech-256). The advantage of the human-computer paradigm is that we can provide usable services to people in the interim-period where computer vision is still unsolved. This may help increase demand for computer vision, spur data collection, and provide solutions for the types of problems people outside the field want solved.

In this work, our goal is to provide a simple framework that makes it as effortless as possible for researchers to plug their existing algorithms into the human-computer framework and use humans to drive up performance to levels that are good enough for real-life applications. Implicit to our model is the assumption that lay-people generally cannot recognize finer-grained categories (*e.g.*, Myrtle Warbler, Thruxton Jackaroo, *etc.*) due to imperfect memory or limited experiences; however, they do have the fundamental visual capabilities to recognize the parts and attributes that collectively make recognition possible (see Fig. 1). By contrast, computers lack many of the fundamental visual capabilities that humans have, but have perfect memory and are able to pool knowledge collected from large groups of people. Users interact with our system by answering simple yes/no or multiple choice questions about an image or object, as shown in Fig. 2. Similar to the *20-questions game*[1], we observe that the

[1] See for example http://20q.net

Fig. 2. Examples of the visual 20 questions game on the 200 class Bird dataset. Human responses (shown in red) to questions posed by the computer (shown in blue) are used to drive up recognition accuracy. In the left image, computer vision algorithms can guess the bird species correctly without any user interaction. In the middle image, computer vision reduces the number of questions to 2. In the right image, computer vision provides little help.

number of questions needed to classify an object from a database of C classes is usually $O(\log C)$ (when user responses are accurate), and can be faster when computer vision is in the loop. Our method of choosing the next question to ask uses an information gain criterion and can deal with noisy (probabilistic) user responses. We show that it is easy to incorporate any computer vision algorithm that can be made to produce a probabilistic output over object classes.

Our experiments in this paper focus on bird species categorization, which we take to be a representative example of recognition of tightly-related categories. The bird dataset contains 200 bird species and over 6,000 images. We believe that similar methodologies will apply to other object domains.

The structure of the paper is as follows: In Section 2, we discuss related work. In Section 3, we define the hybrid human-computer problem and basic algorithm, which includes methodologies for modeling noisy user responses and incorporating computer vision into the framework. We describe our datasets and implementation details in Section 4, and present empirical results in Section 5.

2 Related Work

Recognition of tightly related categories is still an open area in computer vision, although there has been success in a few areas such as book covers and movie posters (*e.g.*, rigid, mostly flat objects [3]). The problem is challenging because the number of object categories is larger, with low interclass variance, and variability in pose, lighting, and background causes high intraclass variance. Ability to exploit domain knowledge and cross-category patterns and similarities becomes increasingly important.

There exist a variety of datasets related to recognition of tightly-related categories, including Oxford Flowers 102 [4], UIUC Birds [5], and STONEFLY9 [6]. While these works represent progress, they still have shortcomings in scaling to large numbers of categories, applying to other types of object domains, or achieving performance levels that are good enough for real-world applications. Perhaps most similar in spirit to our work is the Botanist's Field Guide [7], a system for plant species recognition with hundreds of categories and tens of

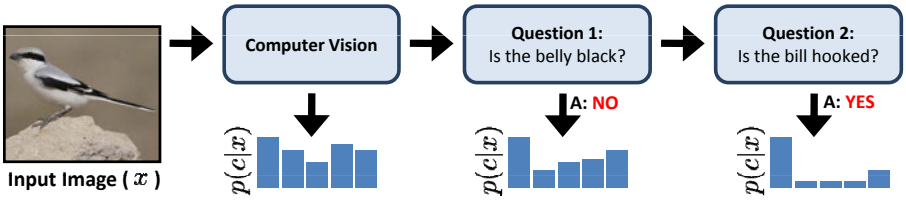

Fig. 3. Visualization of the basic algorithm flow. The system poses questions to the user, which along with computer vision, incrementally refine the probability distribution over classes.

thousands of images. One key difference is that their system is intended primarily for experts, and requires plant leaves to be photographed in a controlled manner at training and test time, making segmentation and pose normalization possible. In contrast, all of our training and testing images are obtained from Flickr in unconstrained settings (see Fig. 4), and the system is intended to be used by lay people.

There exists a multitude of different areas in computer science that interleave vision, learning, or other processing with human input. Relevance feedback [8] is a method for interactive image retrieval, in which users mark the relevance of image search results, which are in turn used to create a refined search query. Active learning algorithms [9,10,11] interleave training a classifier with asking users to label examples, where the objective is to minimize the total number of labeling tasks. Our objectives are somewhat similar, except that we are querying information at runtime rather than training time. Expert systems [12,13] involve construction of a knowledge base and inference rules that can help non-experts solve a problem. Our approach differs due to the added ability to observe image pixels as an additional source of information. Computationally, our method also has similarities to algorithms based on information gain, entropy calculation, and decision trees [14,15,16].

Finally, a lot of progress has been made on trying to scale object recognition to large numbers of categories. Such approaches include using class taxonomies [17,18], feature sharing [19], error correcting output codes (ECOC) [20], and attribute based classification methods [21,22,23]. All of these methods could be easily plugged into our framework to incorporate user interaction.

3 Visual Recognition with Humans in the Loop

Given an image x, our goal is to determine the true object class $c \in \{1...C\}$ by posing questions based on visual properties that are easy for the user to answer (see Fig. 1). At each step, we aim to exploit the visual content of the image and the current history of question responses to intelligently select the next question. The basic algorithm flow is summarized in Fig. 3.

Let $\mathcal{Q} = \{q_1...q_n\}$ be a set of possible questions (*e.g.*, IsRed?, HasStripes?, *etc.*), and $\mathcal{A}_i$ be the set of possible answers to q_i. The user's answer is some

Algorithm 1. Visual 20 Questions Game

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

random variable $a_i \in \mathcal{A}_i$. We also allow users to qualify each response with a confidence value $r_i \in \mathcal{V}$, (*e.g.*, $\mathcal{V} = \{\mathsf{Guessing}, \mathsf{Probably}, \mathsf{Definitely}\}$). The user's response is then a pair of random variables $u_i = (a_i, r_i)$.

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 we will ask the user. $U^{t-1} = \{u_{j(1)}...u_{j(t-1)}\}$ is the set of responses obtained by time step $t-1$. We use maximum information gain as the criterion to select $q_{j(t)}$. Information gain is widely used in decision trees (*e.g.* [15]) and can be computed from an estimate of $p(c|x, U^{t-1})$.

We define $\mathrm{I}(c; u_i | x, U^{t-1})$, the expected information gain of posing the additional question q_i, as follows:

$$\mathrm{I}(c; u_i | x, U^{t-1}) = \mathbb{E}_u \left[\mathrm{KL} \left(p(c|x, u_i \cup U^{t-1}) \parallel p(c|x, U^{t-1}) \right) \right] \quad (1)$$

$$= \sum_{u_i \in \mathcal{A}_i \times \mathcal{V}} p(u_i | x, U^{t-1}) \big(\mathrm{H}(c|x, u_i \cup U^{t-1}) - \mathrm{H}(c|x, U^{t-1}) \big) \quad (2)$$

where $\mathrm{H}(c|x, U^{t-1})$ is the entropy of $p(c|x, U^{t-1})$

$$\mathrm{H}(c|x, U^{t-1}) = -\sum_{c=1}^{C} p(c|x, U^{t-1}) \log p(c|x, U^{t-1}) \quad (3)$$

The general algorithm for interactive object recognition is shown in Algorithm 1. In the next sections, we describe in greater detail methods for modeling user responses and different methods for incorporating computer vision algorithms, which correspond to different ways to estimate $p(c|x, U^{t-1})$.

3.1 Incorporating Computer Vision

When no computer vision is involved it is possible to pre-compute a decision tree that defines which question to ask for every possible sequence of user responses. With computer vision in the loop, however, the best questions depend dynamically on the contents of the image.

In this section, we propose a simple framework for incorporating any multi-class object recognition algorithm that produces a probabilistic output over classes. We can compute $p(c|x, U)$, where U is any arbitrary sequence of responses, as follows:

$$p(c|x, U) = \frac{p(U|c, x) p(c|x)}{Z} = \frac{p(U|c) p(c|x)}{Z} \quad (4)$$

where $Z = \sum_c p(U|c)p(c|x)$. Here, we make the assumption that $p(U|c, x) = p(U|c)$; effectively this assumes that the types of noise or randomness that we see in user responses is class-dependent and not image-dependent. We can still accommodate variation in responses due to user error, subjectivity, external factors, and intraclass variance; however we throw away some image-related information (for example, we lose ability to model a change in the distribution of user responses as a result of a computer-vision-based estimate of object pose).

In terms of computation, we estimate $p(c|x)$ using a classifier trained offline (more details in Section 4.3). Upon receiving an image, we run the classifier once at the beginning of the process, and incrementally update $p(c|x, U)$ by gathering more answers to questions from the user. One could imagine a system where a learning algorithm is invoked several times during the process; as categories are weeded out by answers, the system would use a more tuned classifier to update the estimate of $p(c|x)$. However, our preliminary experiments with such methods did not show an advantage[2]. Note that when no computer vision is involved, we simply replace $p(c|x)$ with a prior $p(c)$.

3.2 Modeling User Responses

Recall that for each question we may also ask a corresponding confidence value from the user, which may be necessary when an attribute cannot be determined (for example, when the associated part(s) are not visible). We assume that the questions are answered independently given the category:

$$p(U^{t-1}|c) = \prod_{i}^{t-1} p(u_i|c) \tag{5}$$

The same assumption allows us to express $p(u_i|x, U^{t-1})$ in Equation 2 as

$$p(u_i|x, U^{t-1}) = \sum_{c=1}^{C} p(u_i|c)p(c|x, U^{t-1}) \tag{6}$$

It may also be possible to use a more sophisticated model in which we estimate a full joint distribution for $p(U^{t-1}|c)$; in our preliminary experiments this approach did not work well due to insufficient training data.

To compute $p(u_i|c) = p(a_i, r_i|c) = p(a_i|r_i, c)p(r_i|c)$, we assume that $p(r_i|c) = p(r_i)$. Next, we compute each $p(a_i|r_i, c)$ as the posterior of a multinomial distribution with Dirichlet prior $\mathrm{Dir}\big(\alpha_r p(a_i|r_i) + \alpha_c p(a_i|c)\big)$, where α_r and α_c are constants, $p(a_i|r_i)$ is a global attribute prior, and $p(a_i|c)$ is estimated by pooling together certainty labels. In practice, we use a larger prior term for *Guessing* than *Definitely*, $\alpha_{guess} > \alpha_{def}$, which effectively down weights the importance of any response with certainty level *Guessing*.

[2] See supplementary material (`http://www.vision.caltech.edu/visipedia/birds200.html`) for more details.

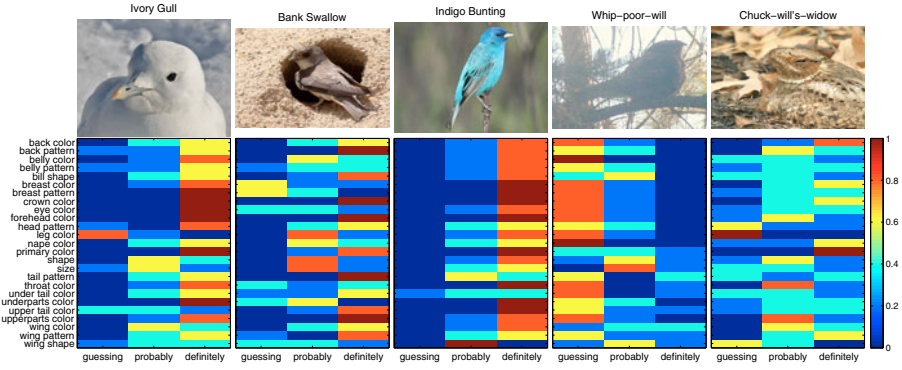

Fig. 4. Examples of user responses for each of the 25 attributes. The distribution over {*Guessing, Probably, Definitely*} is color coded with blue denoting 0% and red denoting 100% of the five answers per image attribute pair.

4 Datasets and Implementation Details

In this section we provide a brief overview of the datasets we used, methods used to construct visual questions, computer vision algorithms we tested, and parameter settings.

4.1 Birds-200 Dataset

Birds-200 is a dataset of 6033 images over 200 bird species, such as Myrtle Warblers, Pomarine Jaegars, and Black-footed Albatrosses – classes that cannot usually be identified by non-experts. In many cases, different bird species are nearly visually identical (see Fig. 8).

We assembled a set of 25 visual questions (list shown in Fig. 4), which encompass 288 binary attributes (*e.g.*, the question HasBellyColor can take on 15 different possible colors). The list of attributes was extracted from whatbird.com[3], a bird field guide website.

We collected "deterministic" class-attributes by parsing attributes from whatbird.com. Additionally, we collected data of how non-expert users respond to attribute questions via a Mechanical Turk interface. To minimize the effects of user subjectivity and error, our interface provides prototypical images of each possible attribute response. The reader is encouraged to look at the supplementary material for screenshots of the question answering user-interface and example images of the dataset.

Fig. 4 shows a visualization of the types of user response results we get on the Birds-200 dataset. It should be noted that the uncertainty of the user responses strongly correlates with the parts that are visible in an image as well as overall difficulty of the corresponding bird species.

[3] `http://www.whatbird.com/`

When evaluating performance, test results are generated by randomly selecting a response returned by an MTurk user for the appropriate test image.

4.2 Animals with Attributes

We also tested performance on the Animals With Attributes (AwA) [21], a dataset of 50 animal classes and 85 binary attributes. We consider this dataset less relevant than birds (because classes are recognizable by non-experts), and therefore do not focus as much on this dataset.

4.3 Implementation Details and Parameter Settings

For both datasets, our computer vision algorithms are based on Andrea Vedaldi's publicly available source code [24], which combines vector-quantized geometric blur and color/gray SIFT features using spatial pyramids, multiple kernel learning, and per-class 1-vs-all SVMs. We added features based on full image color histograms and vector-quantized color histograms. For each classifier we used Platt scaling [25] to learn parameters for $p(c|x)$ on a validation set. We used 15 training examples for each Birds-200 class and 30 training examples for each AwA class. Bird training and testing images are roughly cropped.

Additionally, we compare performance to a second computer vision algorithm based on attribute classifiers, which we train using the same features/training code, with positive and negative examples set using whatbird.com attribute labels. We combined attribute classifiers into per-class probabilities $p(c|x)$ using the method described in [21].

For estimating user response statistics on the Birds-200 dataset, we used $\alpha_{guess} = 64$, $\alpha_{prob} = 16$, $\alpha_{def} = 8$, and $\alpha_c = 8$ (see Section 3.2).

5 Experiments

In this section, we provide experimental results and analysis of the hybrid-human computer classification paradigm. Due to space limitations, our discussion focuses on the Birds dataset. We include results (see Fig. 9) from which the user can verify that trends are similar on Birds-200 and AwA, and we include additional results on AwA in the supplementary material.

5.1 Measuring Performance

We use two main methodologies for measuring performance, which correspond to two different possible user-interfaces:

- **Method 1:** We ask the user exactly T questions, predict the class with highest probability, and measure the percent of the time that we are correct.
- **Method 2:** After asking each question, we present a small gallery of images of the highest probability class, and allow the user to stop the system early. We measure the average number of questions asked per test image.

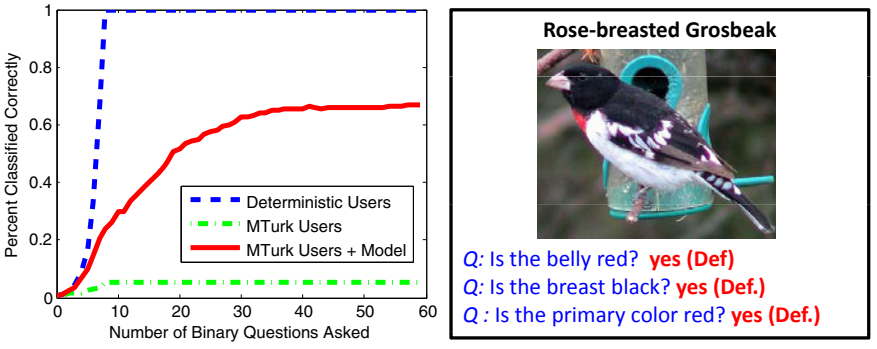

Fig. 5. Different Models of User Responses: *Left:* Classification performance on Birds-200 (Method 1) without computer vision. Performance rises quickly (blue curve) if users respond deterministically according to whatbird.com attributes. MTurk users respond quite differently, resulting in low performance (green curve). A learned model of MTurk responses is much more robust (red curve). *Right:* A test image where users answer several questions incorrectly and our model still classifies the image correctly.

For the second method, we assume that people are perfect verifiers, *e.g.*, they will stop the system if and only if they have been presented with the correct class. While this is not always possible in reality, there is some trade-off between classification accuracy and amount of human labor, and we believe that these two metrics collectively capture the most important considerations.

5.2 Results

In this section, we present our results and discuss some interesting trends toward understanding the visual 20 questions classification paradigm.

User Responses are Stochastic: In Fig. 5, we show the effects of different models of user responses without using any computer vision. When users are assumed to respond deterministically in accordance with the attributes from whatbird.com, performance rises quickly to 100% within 8 questions (roughly $\log_2(200)$). However, this assumption is not realistic; when testing with responses from Mechanical Turk, performance saturates at around 5%. Low performance caused by subjective answers are unavoidable (*e.g.*, perception of the color brown vs. the color buff), and the probability of the correct class drops to zero after any inconsistent response. Although performance is 10 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 Section 3.2) is incorporated, performance jumps to 66% due to the ability to tolerate a reasonable degree of error in user responses (see Fig. 5 for an example). Nevertheless, stochastic user responses increase the number of questions required to achieve a given accuracy level, and some images can never be classified correctly, even when asking all possible questions. In Section 5.2, we discuss the reasons why performance saturates at lower than 100% performance.

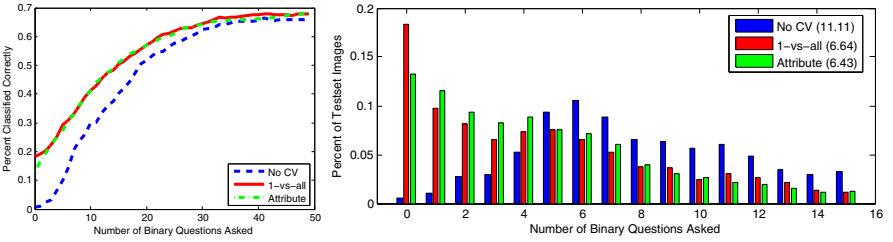

Fig. 6. Performance on Birds-200 when using computer vision: Left Plot: comparison of classification accuracy (Method 1) with and without computer vision when using MTurk user responses. Two different computer vision algorithms are shown, one based on per-class 1-vs-all classifiers and another based on attribute classifiers. Right plot: the number of questions needed to identify the true class (Method 2) drops from 11.11 to 6.43 on average when incorporating computer vision.

Computer Vision Reduces Manual Labor: The main benefit of computer vision occurs due to reduction in human labor (in terms of the number of questions a user has to answer). In Fig. 6, we see that computer vision reduces the average number of yes/no questions needed to identify the true bird species from 11.11 to 6.43 using responses from MTurk users. Without computer vision, the distribution of question counts is bell-shaped and centered around 6 questions. When computer vision is incorporated, the distribution peaks at 0 questions but is more heavy-tailed, which suggests that computer vision algorithms are often good at recognizing the "easy" test examples (examples that are sufficiently similar to the training data), but provide diminishing returns toward classifying the harder examples that are not sufficiently similar to training data. As a result, computer vision is more effective at reducing the average amount of time than reducing the time spent on the most difficult images.

User Responses Drive Up Performance: An alternative way of interpreting the results is that user responses drive up the accuracy of computer vision algorithms. In Fig. 6, we see that user responses improve overall performance from $\approx 19\%$ (using 0 questions) to $\approx 66\%$.

Computer Vision Improves Overall Performance: Even when users answer all questions, performance saturates at a higher level when using computer vision ($\approx 69\%$ vs. $\approx 66\%$, see Fig. 6). The left image in Fig. 7 shows an example of an image classified correctly using computer vision, which is not classified correctly without computer vision, even after asking 60 questions. In this example, some visually salient features like the long neck are not captured in our list of visual attribute questions. The features used by our vision algorithms also capture other cues (such as global texture statistics) that are not well-represented in our list of attributes (which capture mostly color and part-localized patterns).

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 possible classes roughly in half. Questions

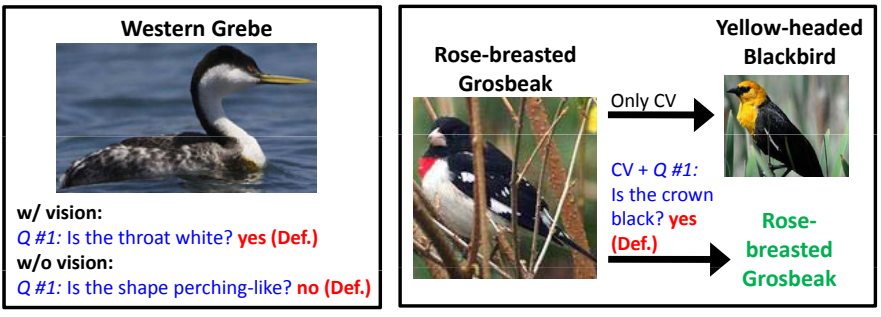

Fig. 7. Examples where computer vision and user responses work together: *Left:* An image that is only classified correctly when computer vision is incorporated. Additionally, the computer vision based method selects the question HasThroatColorWhite, a different and more relevant question than when vision is not used. In the right image, the user response to HasCrownColorBlack helps correct computer vision when its initial prediction is wrong.

like HasShapePerchingLike, which divide the classes fairly evenly, and HasUnderpartsColorYellow, which tends to be answered reliably, are commonly chosen.

When computer vision is incorporated, the likelihood of classes change and different questions are selected. In the left image of Fig. 7, 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.

Recognition is Not Always Successful: According the the Cornell Ornithology Website[4], 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 paradigm is that other contextual sources of information such as behavior and habitat can easily be incorporated as additional questions.

Fig. 8 illustrates some example failures. The most common failure conditions occur due to 1) classes that are nearly visually identical, 2) images of poor viewpoint or low resolution, such that some parts are not visible, 3) significant mistakes made by MTurkers, or 4) inadequacies in the set of attributes we used.

1-vs-all Vs. Attribute-Based Classification: In general, 1-vs-all classifiers slightly outperform attribute-based classifiers; however, they converge to similar performance as the number of question increases, as shown in Fig. 6 and 9. The features we use (kernelized and based on bag-of-words) may not be well suited to the types of attributes we are using, which tend to be localized and associated with a particular part. One potential advantage of attribute-based methods is computational scalability when the number of classes increases; whereas 1-vs-all methods always require C classifiers, the number of attribute classifiers can

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

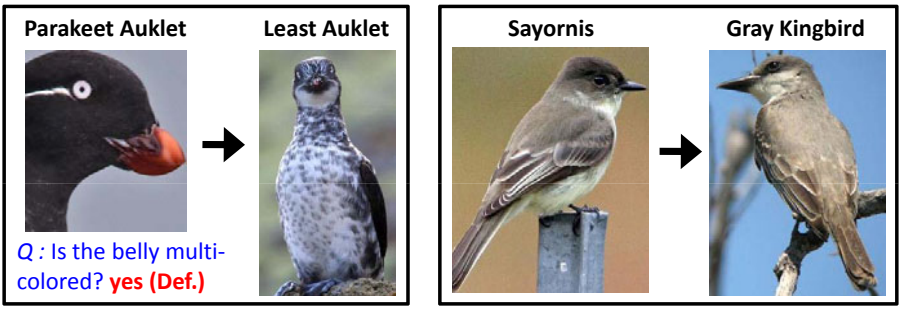

Fig. 8. Images that are misclassified by our system: *Left:* The Parakeet Auklet image is misclassified due to a cropped image, which causes an incorrect answer to the belly pattern question (the Parakeet Auklet has a plain, white belly, see Fig. 2). *Right:* The Sayornis and Gray Kingbird are commonly confused due to visual similarity.

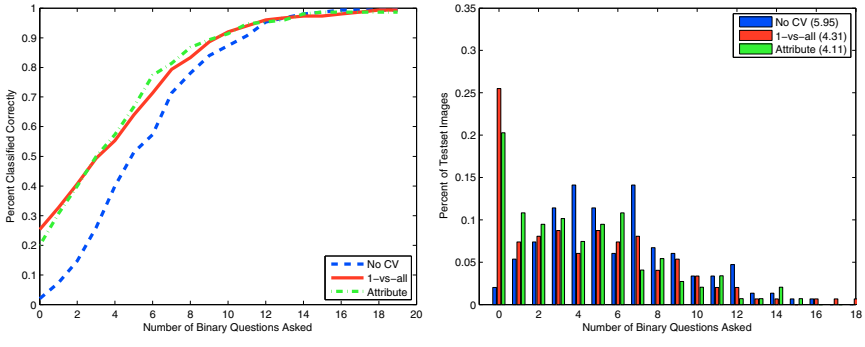

Fig. 9. Performance on Animals With Attributes: Left Plot: Classification performance (Method 1), simulating user responses using soft class-attributes (see [21]). Right Plot: The required number of questions needed to identify the true class (Method 2) drops from 5.94 to 4.11 on average when incorporating computer vision.

be varied in order to trade-off accuracy and computation time. The table below displays the average number of questions needed (Method 1) on the Birds dataset using different number of attribute classifiers (which were selected randomly):

200 (1-vs-all)	288 attr.	100 attr.	50 attr.	20 attr.	10 attr.
6.43	6.72	7.01	7.67	8.81	9.52

6 Conclusion

Object recognition remains a challenging problem for computer vision. Furthermore, recognizing tightly related categories in one shot is difficult even for humans without proper expertise. Our work attempts to leverage the power of both

human recognition abilities and that of computer vision. We presented a simple way of designing a hybrid human-computer classification system, which can be used in conjunction with a large variety of computer vision algorithms. Our results show that user input significantly drives up performance; while it may take many years before object recognition algorithms achieve reasonable performance on their own, incorporating human input can produce usable recognition systems. On the other hand, having computer vision in the loop reduces the amount of required human labor to successfully classify an image. Finally, we showed that incorporating models of stochastic user responses leads to much better reliability in comparison to deterministic field guides generated by experts.

We believe our work opens the door to many interesting sub-problems. The most obvious next step is to explore other types of domains. While we were able to extract a set of reasonable attributes/questions for the bird dataset, this may be more difficult for other domains; one possible topic for future work is to find a more principled way of discovering a set of useful questions. Alternative types of user input, such as asking the user to click on the location of certain parts, could also be investigated. Lastly, while we used off-the-shelf computer vision algorithms in this work, it may be possible to improve them to better suit the challenges of tightly-related category recognition, such as algorithms that incorporate a part-based model.

Acknowledgments

Funding for this work was provided by NSF CAREER Grant #0448615, NSF Grant AGS-0941760, ONR MURI Grant N00014-06-1-0734, ONR MURI Grant #N00014-08-1-0638, Google Research Award. The authors would like to give special thanks to Takeshi Mita for his efforts in constructing the birds dataset.

References

1. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL VOC Challenge 2009 Results (2009)
3. Nister, D., Stewenius, H.: Recognition with a vocabulary tree. In: CVPR (2006)
4. Nilsback, M., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conf. on Comp. Vision, Graphics & Image Proc., pp. 722–729 (2008)
5. Lazebnik, S., Schmid, C., Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In: ICCV, vol. 1, pp. 832–838 (2005)
6. Martınez-Munoz, et al.: Dictionary-free categorization of very similar objects via stacked evidence trees. In: CVPR (2009)
7. Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., Zhang, L.: Searching the world's herbaria: A system for visual identification of plant species. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 116–129. Springer, Heidelberg (2008)

8. Zhou, X., Huang, T.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Systems 8, 536–544 (2003)
9. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. JMLR 2, 45–66 (2002)
10. Kapoor, A., Grauman, K., Urtasun, R., Darrell, T.: Active learning with gaussian processes for object categorization. In: ICCV, pp. 1–8 (2007)
11. Holub, A., Perona, P., Burl, M.: Entropy-based active learning for object recognition. In: Workshop on Online Learning for Classification (OLC), pp. 1–8 (2008)
12. Neapolitan, R.E.: Probabilistic reasoning in expert systems: theory and algorithms. John Wiley & Sons, Inc., New York (1990)
13. Beynon, M., Cosker, D., Marshall, D.: An expert system for multi-criteria decision making using Dempster Shafer theory. Expert Systems with Applications 20 (2001)
14. Tsang, S., Kao, B., Yip, K., Ho, W., Lee, S.: Decision trees for uncertain data. In: International Conference on Data Engineering, ICDE (2009)
15. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
16. Dembo, A., Cover, T., Thomas, J.: Information theoretic inequalities. IEEE Transactions on Information Theory 37, 1501–1518 (1991)
17. Sivic, J., Russell, B., Zisserman, A., Freeman, W., Efros, A.: Unsupervised discovery of visual object class hierarchies. In: CVPR, pp. 1–8 (2008)
18. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: CVPR, pp. 1–8 (2008)
19. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR, vol. 2 (2004)
20. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
21. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)
22. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
23. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and Simile Classifiers for Face Verification. In: ICCV (2009)
24. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV (2009)
25. Platt, J.: Probabilities for SV machines. In: NIPS, pp. 61–74 (1999)