COCO Attributes: Attributes for People, Animals, and Objects

Genevieve Patterson[1(✉)] and James Hays[2]

[1] Microsoft Research, Cambridge, USA
gen@microsoft.com
[2] Georgia Institute of Technology, Atlanta, USA
hays@gatech.edu

Abstract. In this paper, we discover and annotate visual attributes for the COCO dataset. With the goal of enabling deeper object understanding, we deliver the largest attribute dataset to date. Using our COCO Attributes dataset, a fine-tuned classification system can do more than recognize object categories – for example, rendering multi-label classifications such as "sleeping spotted curled-up cat" instead of simply "cat". To overcome the expense of annotating thousands of COCO object instances with hundreds of attributes, we present an Economic Labeling Algorithm (ELA) which intelligently generates crowd labeling tasks based on correlations between attributes. The ELA offers a substantial reduction in labeling cost while largely maintaining attribute density and variety. Currently, we have collected 3.5 million object-attribute pair annotations describing 180 thousand different objects. We demonstrate that our efficiently labeled training data can be used to produce classifiers of similar discriminative ability as classifiers created using exhaustively labeled ground truth. Finally, we provide baseline performance analysis for object attribute recognition.

Keywords: Dataset creation · Attributes · Crowdsourcing · Multilabel recognition

1 Introduction

Traditionally, computer vision algorithms describe objects by giving each instance a categorical label (e.g. cat, Barack Obama, bedroom, etc.). However, category labels provide a limited approximation of the human understanding of natural images. This categorical model has some significant limitations: (1) We have no way to express *intra*-category variations, e.g. "fresh apple" vs. "rotten apple." (2) A categorical representation alone cannot help us to understand the state of objects relative to other objects in a scene. For example, if there are two people arguing in a scene, knowing that they are both 'people' won't help us understand who is angry or who is guilty. (3) The categorical model prevents researchers from responding to complex questions about the contents of a

B. Leibe et al. (Eds.): ECCV 2016, Part VI, LNCS 9910, pp. 85–100, 2016.
DOI: 10.1007/978-3-319-46466-4_6

natural scene. This final limitation is a particular obstacle in Visual Question Answering [1] or the Visual Turing Test [2].

To alleviate these limitations, we aim to add semantic visual attributes [3,4] to objects. The space of attributes is effectively infinite but the majority of possible attributes (e.g., "This man's name is John.", "This book has historical significance.") are not interesting to us. We are interested in finding attributes that are likely to visually distinguish objects from each other (not necessarily along categorical boundaries). In this paper, we expand on the type of attributes introduced by Farhadi et al.

Fig. 1. *Examples from COCO Attributes.* In the figure above, images from the COCO dataset are shown with one object outlined in white. Under the image, the COCO object label is listed on the left, and the COCO Attribute labels are listed on the right. The COCO Attributes labels give a rich and detailed description of the context of the object.

Outline: In Sect. 3, we explain how we determine which attributes to include in our dataset. To determine the attribute taxonomy for COCO, we implement a crowd-in-the-loop content generation system. Section 4 illustrates the burden of taking a naíve approach to attribute labeling. In that section we exhaustively label all of our discovered attributes for a subset of 6500 object instances. This 'exhaustive' sub-dataset is then used to bootstrap our economic labeling pipeline described in Sect. 5. Section 6 presents some baseline classification results on COCO Attributes.

The COCO dataset contains 500,000 images and 2M individually annotated objects. Given the scale of this dataset, it is economically infeasible to annotate all attributes for all object instances. The Economic Labeling Algorithm (ELA) introduced in Sect. 5 approximates the exhaustive annotation process. The ELA selects a subset of attributes that is likely to contain all of the positive labels for a novel image. By labeling the attributes most likely to be positive first, we are able to reduce the number of annotations required without greatly sacrificing overall label recall. We annotate objects from 29 of the most-populated COCO object categories with nearly 200 discovered attributes.

Currently, our COCO Attributes dataset comprises 84,044 images, 188,426 object instances, 196 object attributes, and 3,455,201 object-attribute annotation pairs. The objects in the dataset vary widely, from cars to sandwiches to cats and dogs. In Sect. 3 we employ proven techniques, such as text-mining, image

comparison tasks, and crowd shepherding, to find the 196 attributes we later use to label the dataset [5–9].

Our contribution is straightforward — we obtain attribute labels for thousands of object instances at a reasonable cost. For the sake of estimating an upper bound on the cost of annotating attributes across the COCO dataset, let us assume several figures relating to the number of annotations and cost per annotation using the widely employed Amazon Mechanical Turk platform (AMT).

Let's assume that crowd workers are asked to annotate 50 images per human intelligence task (HIT). Our dataset contains approximately 200 visual attributes. For COCO Attributes, we annotate attributes for a subset of the total COCO dataset, approximately 180,000 objects across 29 object categories. The cost of exhaustively labeling 200 attributes for all of the object instances contained in our dataset would be: 180k objects $\times$ 200 attributes)/50 images per HIT $\times$ (\$0.07 pay per HIT + \$0.014 Amazon fee) = \$60,480. If we annotate each attribute for the top 10 % of object instances mostly likely to contain a particular attribute, the overall annotation cost would drop to a reasonable \$6,048. But how do we discover the most informative and characteristic attributes for each object in the COCO dataset? We present our answer to this question in Sect. 5.

To verify the quality of the COCO Attributes dataset, we explore attribute classification. In Sect. 6, we show that a CNN finetuned on our ELA labeled training set to predict multi-label attribute vectors performs similarly to classifiers trained on exhaustively labeled instances.

2 Related Work

To our knowledge, no attribute dataset has been collected containing both the number of images and the number of object attributes as our COCO Attributes dataset. Existing attribute datasets concentrate on either a small range of object categories or a small number of attributes.

One notable exception is the Visual Genome dataset introduced in Krishna et al. [10], which also aims to provide a dataset of complex real-world interactions between objects and attributes. Visual Genome contains myriad types of annotations, all of which are important for deeper image understanding. For COCO Attributes, we focus on making the largest attribute dataset we possibly can. In that regard we have been able to collect more than double the number of object-attribute pair annotations. COCO Attributes and the Visual Genome dataset together open up new avenues of research in the vision community by providing non-overlapping attribute datasets. Creating COCO Attributes is an experiment in economically scalling up attribute annotation as demonstrated in attribute literature such as the CUB 200 dataset [11], the SUN Attribute dataset [8], Visual Genome, and other well-cited works of attribute annotation [12,13].

Initial efforts to investigate attributes involved labeling images of animals with texture, part, and affordance attributes [3,4,14]. These attributes were

chosen by the researchers themselves, as the interesting attributes for animals were clear at the time of publication. COCO dataset images are more complicated than those in Farhadi et al. [14]. They often have multiple objects, object occlusions, and complicated backgrounds. Attributes are necessary to differentiate objects in COCO scenes or describe the variety in instances of the same category. As a result, the COCO Attributes are more detailed and descriptive than those in earlier datasets.

Other attribute datasets have concentrated on attributes relating to people. Kumar et al. [15] and Liu et al. [16] introduced datasets with face attributes and human activity affordances, respectively. The influential Poselets dataset [17] labeled human poses and has been crucial for the advancement of both human pose and attribute estimation.

Vedaldi et al. [18] used a specialized resource for collecting attributes. They collected a set of discriminative attributes for airplanes by consulting hobbyist and expert interest websites. It is possible that the best method for collecting high-quality attributes is to use a more sophisticated crowd, reserving the general-expertise crowd to label the dataset. We explore bootstrapping the attribute discovery process by mining discriminative words from a corpus of descriptive text written by language 'experts' — novels and newspapers. Similar methods have been demonstrated successfully in [5,12,13]. We use the descriptive words found in these texts to seed a crowd pipeline that winnows the large variety of seed words down to the attributes that visually describe the COCO objects.

Several datasets have collected attributes for the purpose of making visual search more tractable. The Whittlesearch [19] dataset contains 14,658 shoe images with 10 instance-level relative attributes. Parikh and Grauman [20] show that predicted attributes can be used to better describe the relative differences between objects of the same and different categories. This paper furthers the attribute annotation and recognition research begun in those papers by concentrating on scaling up the size of the attribute dataset.

A number of past projects sought to bootstrap dataset annotation using active learning [21–24]. The ELA method presented in Sect. 5 takes a different approach. The ELA is also iterative, exploiting correlation and information gain in a partially labeled training set, but does not use an intermediate classifier. The ELA uses no visual classification.

Vijayanarasimhan and Grauman [21] and Patterson et al. [24] show that the crowd in combination with active learning can rapidly converge on a visual phenomena. However, these active learning systems may be missing the most visually unusual examples. While we seek to annotate COCO Attributes with maximum efficiency, we choose not to make the visual approximation inherently imposed by an active learning pipeline.

Admittedly, our Efficient Labeling Algorithm (ELA) has the possible bias that may occur when we label a subset of the total number of attributes. Section 5 describes the trade-offs among visual diversity, label accuracy, and annotation cost.

We identify additional cost-saving annotation strategies by imitating successes in multi-class recognition [25,26]. Deng et al. define the Hierarchy and Exclusion (HEX) graph, which captures semantic relationships between object labels [26]. HEX graphs describe whether a pair of labels are mutually exclusive, overlap, or subsume one or the other. In Deng et al. HEX graphs are used for object classification. We use the hierarchy of the COCO objects to inform our economic labeling algorithm (ELA), described in Sect. 5.

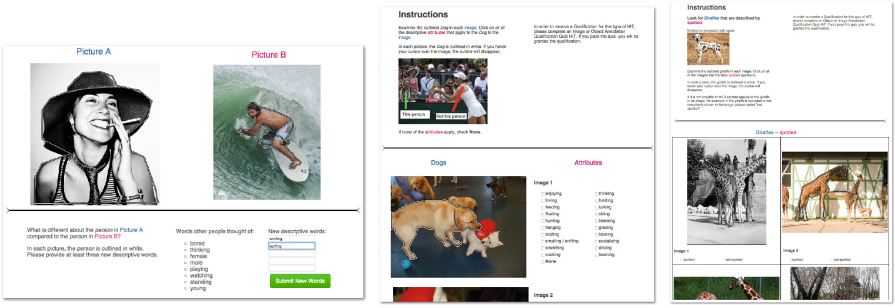
This paper introduces a large new dataset and a novel way to cheaply collect annotations without introducing visual bias. In Sect. 3, we use the crowd to curate our collection of attributes. Section 4 presents a baseline for exhaustive annotation. Section 5 introduces the ELA and shows how we improve on the cost of the exhaustive baseline. Section 5 demonstrates that a dataset collected with the ELA protocol has similar annotation density to an exhaustively labeled dataset. Section 6 contrasts classifiers trained on ELA generated labels and exhaustive labels to show that labeling bias is minimal. Finally, Sect. 6 compares recognizing attributes individually or in a multilabel setting.

3 Attribute Discovery

The first stage of creating the COCO Attributes dataset is determining a taxonomy of relevant attributes. For COCO Attributes, we search for attributes for all of the object categories contained under the COCO super-categories of Person, Vehicle, Animal, and Food. These categories are person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, bird, cat, dog, horse, cow, sheep, elephant, bear, zebra, giraffe, banana, apple, orange, broccoli, carrot, hot dog, pizza, donut, cake, and sandwich. We must determine the attributes that would be useful for describing these objects.

When annotating the COCO objects, we will use a universal taxonomy of attributes versus a category specific taxonomy. Objects from all categories will be annotated with all attributes. Of course, some attributes won't occur at all for certain categories (we didn't observe any "furry" cars), but other attributes like "shiny" manifest across many categories. Certain attributes may have very different visual manifestation in different categories, e.g. an "saucy" pizza and an "saucy" person don't necessarily share the same visual features. We aim to find a large corpus of attributes that will describe both specific categories and often be applicable to several unrelated categories.

Asking Amazon Mechanical Turk (AMT) workers to describe the objects from scratch might result in terms that do not generalize well to other objects in the same hierarchical group or are too common to be discriminative, for example 'orange' does not help us describe the difference between oranges. To bootstrap the attribute discovery process, we mine a source of English text likely to contain descriptive words – the New York Times Annotated Corpus [27]. This corpus contains all of the articles published by the NYT from 1987–2008. We extract all adjectives and verbs occurring within five words of one of our object words. This results in hundreds of descriptive words. Unfortunately, not all of these candidate attributes describe visually recognizable phenomena.



(a) *Attribute Discovery User Interface (UI).*

(b) *Exhaustive Annotation UI.*

(c) *Economic Annotation UI.*

Fig. 2. Amazon Mechanical Turk (AMT) Task Interfaces used in the creation of COCO Attributes.

In order to filter the attributes mined from the NYT corpus, and indeed add a few new ones, we design an AMT Human Intelligence Task (HIT). Our attribute discovery HIT, shown in Fig. 2a, encourages AMT workers to submit visual attributes by asking them to discriminate between two images. In this experiment, we show workers two randomly selected COCO objects from the same category. The worker types in several words that describe one of the images but not the other. To help focus our workers and guide them to make better suggestions, we show a random subsampling of the attributes discovered via the NYT corpus or submitted in previous HITs.

In the end, approximately 300 unique terms were submitted by AMT workers to describe the 29 different categories. The authors manually condensed the combined list of NYT corpus attributes and AMT worker attributes. Attributes that do not refer to a visual property were also removed, e.g. 'stolen' or 'unethical'. The final attribute list comprises 196 attributes.

4 Exhaustive Annotation

In annotating our attributes we would like to avoid asking redundant questions (e.g. asking if a person is "sitting" when they're already labeled as "standing"). To intelligently avoid these situations we need to understand the correlations among attributes. We first build an exhaustively annotated dataset that has a ground truth label obtained via the crowd for every possible object-attribute pair. Our exhaustively labeled dataset serves as a training set for the ELA method we will introduce in Sect. 5. A portion of the exhaustively labeled set is set aside as a validation set to measure the performance of the ELA.

To create the exhaustively labeled part of the COCO Attributes dataset, we employ the annotation UI shown in Fig. 2b for AMT. The object instances in this part of the dataset were chosen as follows: for all categories we exhaustively annotate 10 % of object instances that are larger than 32×32 px. AMT workers

are shown 10 images per HIT and 20 possible attributes subsampled from the total 196. Workers are asked to check all attributes that apply to the object outlined in white. The attributes are roughly grouped by type, such as action word, emotion, surface property, etc.

To improve annotation quality, we implement several quality control techniques. We require that workers complete an annotation quiz in order to begin working on HITs. The quiz looks identical to the UI in Fig. 2b. The worker is required to score 90 % recall of the attributes present in that HIT. Labels are repeated by three workers in order to establish a consensus value for each label. Workers are discarded if their work is checked by the authors and found to be poor. The authors completed $>$ 4,000 annotations. Workers are flagged for a check when they disagree too frequently with the authors' or trusted worker annotations. We define "too frequently" as disagreeing more often than a standard deviation away from the average disagreement of trusted workers. Trusted workers are established by the author's manual review.

We annotate a total of 20,112 object instances with all 196 attributes (5000 person instances and approximately 500 instances of every other object). If two of the three annotations for an attribute are positive, we consider it a true positive.

Responding to comments from our workers, we pay \$0.10 per exhaustive annotation HIT. In total, this portion of the dataset cost: 20112 images $\times$ 196 attributes)/avg. 196 annotations per HIT $\times$ (\$0.10 pay per HIT + \$0.02 Amazon fee) $\times$ 3 workers repeat each annotation $\approx$ \$7,240. If we continued this annotation policy to annotate the remaining 'person', 'animal', 'vehicle', and 'food' objects from COCO (285k instances), the total annotation cost would be **\$102,600**. Using the ELA, we will be able to accomplish this task for only **\$26,712**. That is the price of labeling the remaining 265 K object instances, querying 10 % of the attributes, and using our ELA MTurk HIT that shows 50 images per task (265k object $\times$ 20 attributes/50 images per HIT $\times$ \$0.084 Amazon Fee $\times$ 3 workers = \$26,712).

5 Economic Labeling

Attributes in many domains are sparse. With 196 attributes, we find that across all 29 object categories, the average number of positive attributes per object is 9.35. Ideally, we could identify the most likely attributes that are positive for each object and only ask the AMT workers to annotate (or verify) those attributes. Annotating a new dataset with a huge number of possible attributes would then be relatively inexpensive. Unfortunately, we do not possess an oracle capable of identifying the perfect set of attributes to ask about.

Without the benefit of an attribute oracle, we apply a method of selecting attributes that are likely to be positive for a given object instance. We begin with the set of COCO Attributes A. For an unlabeled object, we calculate the probability $P(a_i = 1|y)$ that an attribute $a_i \in A$ is true given the category y. Equation (3) calculates this likelihood as the mean of the probability of that attribute in all observations of the object category $\mathcal{I}_y$ (Eq. 1) and the probability

of that attribute in the sibling object categories x that are part of the same super category S as y (Eq. 2). The object super categories and their child relationships are defined as part of the COCO dataset.

$$P(a_i = 1|\mathcal{I}_y) = \frac{N^1_{a_i,\mathcal{I}_y}}{N_{\mathcal{I}_y}} \tag{1}$$

To avoid a zero count for a rare attribute for object category y, we count the occurrences of a_i in all instances $\mathcal{I}_x$ of the sibling categories x of y, for $x \in S$. The hierarchical super-category S contains K sub-categories.

$$P(a_i = 1|\mathcal{I}_S) = \frac{\sum_{x=1}^{K} N^1_{a_i,\mathcal{I}_x}}{\sum_{x=1}^{K} N_{\mathcal{I}_x}} \tag{2}$$

We calculate the probability of a_i given y by the average of $P(a_i = 1|\mathcal{I}_y)$ and $P(a_i = 1|\mathcal{I}_S)$.

$$P(a_i = 1|y) = \frac{1}{2} * (P(a_i = 1|\mathcal{I}_y) + P(a_i = 1|\mathcal{I}_S)) \tag{3}$$

For example, Eq. 3 would calculate the probability of 'glazed' given 'donut' by calculating the percent of images where 'glazed' and 'donut' or 'glazed' and <other food category> were present for the same object in the set of exhaustively labeled examples. If an attribute such as 'dirty' never co-occurs with 'horse', Eq. 3 uses just the percent of images where 'dirty' co-occurred with any other animal category like 'cat' and 'dog'.

To select the most likely attribute for the object instance $\hat{a}$, given the set all of the labeled instances of category $\mathcal{I}_y$, we use by Eq. (4).

$$\hat{a} = \arg\max_{a_i \in A} P(a_i = 1|y) \tag{4}$$

Essentially, our economic labeling algorithm follows these steps: (1) Obtain an exhaustively annotated training set $\mathcal{T}$. (2) For each object instance $\mathcal{I}_j$ in the unlabed dataset $\mathcal{D}$, label the most likely attribute from $\mathcal{T}$ calculated using Eq. (3). (3) Select the subset of labeled object instances from $\mathcal{T}$ that share the object category y of $\mathcal{I}_j$. (4) Annotate the attribute $\hat{a}$ (Eq. (4)) for object $\mathcal{I}_j$. (5) Repeat this process until each object in $\mathcal{D}$ has at least N attributes labeled, resulting in the labeled attribute dataset $\mathcal{D}'$. After the first round of labeling, step (3) is slightly changed so that y represents all object instances of category y that also share the attributes labeled in the previous iterations $\hat{a}_{n-1}$. This process is more precisely described in Algorithm 1.

There is one stage of the ELA that presents a tricky problem. What should be done if the subset of $\mathcal{T}$ returned by the function `MatchingSubset` in Algorithm 1 is empty? We explore four possible alternatives for overcoming the problem of an uninformative subset.

Our four alternative varieties of the `AltMatchingMethod` method are only used if either the matching subset is empty or the remaining attributes from

Input: Dataset $\mathcal{D}$ of unlabeled images, fully labeled training set $\mathcal{T}$, labels to annotate A
Output: Labeled dataset $\mathcal{D}'$

```
for I_j ∈ D do
    ▷ I_j is an unlabeled image from D
    while NumLabels(I_j) < N do
        ▷ Repeat annotation until N labels are acquired
        D_S = MatchingSubset(I_j, D)
        if isEmpty(D_S) then
            D_S = AltMatchingMethod(I_j, D)
        end
        Q_n = SelectAttributeQuery(D_S)
        I_j[n] = Annotate(Q_n)
    end
end
return D'
```

Algorithm 1. Economic Labeling Algorithm (ELA)

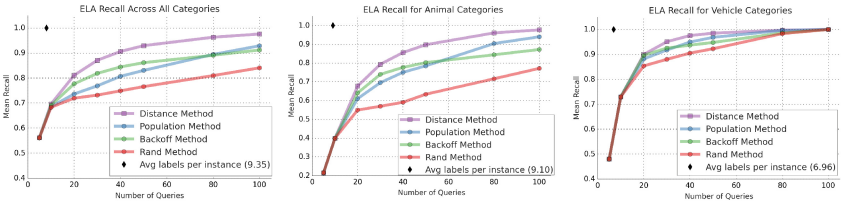
the matching subset have no positive labels. Otherwise, the ELA continues to ask for annotation of the most popular attributes in decreasing order. For our experiments, we also deemed a matching subset to be "empty" if it contained fewer than 5 matching instances.

The first alternative is the Random method, which randomly selects the next attribute to label from the set of unlabeled attributes. The second strategy is the Population method, which proceeds to query the next most popular attribute calculated from the whole labeled set $\mathcal{T}$.

Our third alternative, Backoff, retreats backward through the previously calculated subsets until a subset with a positive attribute that has not been labeled is found. For example, if a dog instance is annotated first with 'standing' and then with 'not furry', there may be no matching dog instances in the training that have both of those labels. The Backoff method would take the subset of training instances labeled 'dog' and 'standing', calculate the second most popular attribute after 'furry', and ask about that second most popular attribute. The Backoff method is similar to the Population method except that the Population method effectively backs off all the way to the beginning of the decision pipeline to decide the next most popular attribute.

The fourth method we explore is the Distance method. This alternative uses the current subset of annotated attributes as a feature vector and finds the 100 nearest neighbors from the set $\mathcal{T}$, given only the subset of currently labeled attributes. For example, if a partially labeled example had 10 attributes labeled, the nearest neighbors would be calculated using the corresponding 10-dimensional feature vector. The next most popular attribute is selected from the set of nearest neighbors.

In order to compare these alternative methods, we split our exhaustively labeled dataset into test and train sets. We use 19k object instances for the training set and 1k object instances for test. The object instances are randomly selected from all 29 object categories. In our simulation, we use the ELA methods to generate the annotated attributes for the test set. For each object instance in the test set, we begin by knowing the category and super-category labels. For example, given a test image we might know that it is a 'dog' and an 'animal'.



(a) All Categories (b) Animals (c) Vehicles

Fig. 3. *Mean Recall Comparison of Alternative ELA methods.* (a) plots the mean recall of the test dataset alternatively labeled with each ELA method and stopped for a range of query limits. All categories were included in this comparison. The Distance method is the clear winner, obtaining 80 % recall for only 20 attribute queries, approximately 10 % of the total number of attributes. (b) and (c) show the mean recall across all test instances of their type of category. The vehicle categories achieve a higher recall with fewer queries than the animal categories. This may be due to the smaller subset of attributes relevant to vehicles than to animals.

We proceed by following the steps of the ELA up to a limit of N attribute queries. We determine the response to each query by taking the ground truth value from the test set. In this way we simulate 3 AMT workers responding to an attribute annotation query and taking their consensus. After N queries, we calculate the recall for that instance by comparing the number of positive attribute annotations in the ELA label vector to the exhaustive label vector. If we use the ELA to label a 'dog' instance up to 20 queries and obtain 8 positive attributes, but the ground truth attribute vector for that dog has 10 positive attributes, then the recall for that instance is 0.8.

We compare the four alternative methods in Fig. 3. Each method is tested on 1k object instances, and the mean recall score averages the recall of all test instances. In all of the plots in Fig. 3, the four methods preform approximately the same for the first 10 attribute queries. This is to be expected as none of the methods will perform differently until the partially labeled instances become sufficiently distinctive to have no matching subsets in the training set.

After approximately 10 queries, the methods begin to diverge. The Random method shows linear improvement with the number of queries. If these plots were extrapolated to 196 queries, the Random method would achieve a recall of 1.0 at query 196. The other methods improve faster, approaching perfect recall much sooner. The Backoff method initially out-performs the population method, indicating that at early stages querying the popular attributes from a more specific subset is a better choice than querying the most popular attributes overall. This distinction fails to be significant after more queries are answered.

The best performing method is the Distance method. This method comes closer to selecting the best subset of attribute queries both for the full hierarchy and for the animal and vehicle sub-trees. The success of the Distance method indicates that the most likely next attribute for a given image may be found by looking at examples that are similar but not exactly the same as a given object.

Based on the Distance method results in Fig. 3, we use a different number of attribute queries per object type to annotate COCO Attributes. We ask enough questions to ensure that each object type's attributes have mean recall of 80 %, e.g. 30 for animals, 17 for vehicles, etc. We ask for 20 attributes on average according to Fig. 3a.

To further examine the performance of the ELA with the Distance method, we plot a selection of per attribute recall scores in Fig. 4. The attributes in Fig. 4 are sorted by ascending population in the dataset. One would expect the more popular attributes to have higher recall than the less popular attributes for a lower number of attribute queries. This is not strictly the case however. 'Stretching', for example, is a popular attribute, but does not obtain higher than 0.9 recall until 100 queries. This indicates that 'stretching' is not strongly correlated with other attributes in the dataset. Conversely, even at 20 attribute queries many of the rarer attributes still have a reasonable chance of being queried.

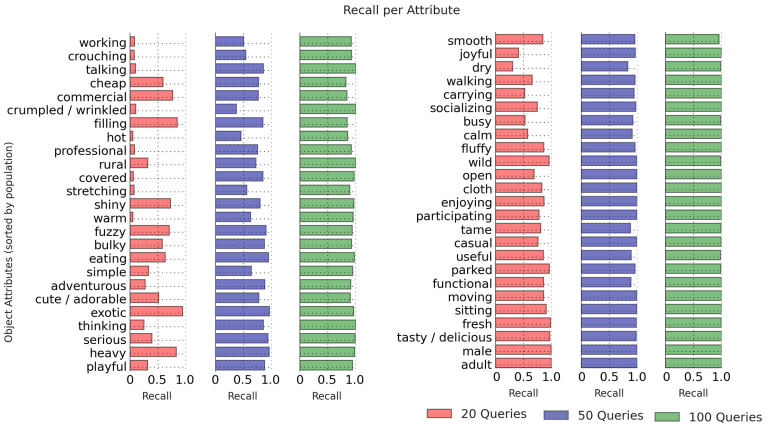


Fig. 4. *Mean Recall Across all Categories for 50 Attributes.* This plot shows the recall of the ELA-Distance method for annotating 50 randomly selected attributes from the full set of 196. The recall is calculated across all instances of the exhaustively labeled test set. The attributes are sorted by their popularity in the exhaustively labeled dataset. Mean chance recall is 0.092, and varies from 0.03 to 0.1 across attributes. The ELA beats chance recall for all attributes.

We also attempt hybrid versions of the methods described above. We repeat the simulation shown in Fig. 3 by first annotating the top 10 most popular attributes, and then continuing with the alternate methods. In this way we might be able to to discover unusual objects early, thus making our method more robust. However, the performance of the hybrid methods were barely different than that shown in Fig. 3.

Figure 4 shows where the ELA does cause a bias by missing the "tail" of rare attributes with too few attribute queries. But this is not a *visual* bias linked

Fig. 5. *Examples from COCO Attributes.* These are positive examples of the listed attributes from the dataset. Examples such as the man cuddling a horse or the dog riding a surfboard shows how this dataset adds important context to image that would otherwise be lost by only listing the objects present in an image.

to a particular feature or classifier as would be the case with an active learning approach. It would be problematic if dataset errors were linked to a particular feature and future approaches were implicitly penalized for being different from that feature. The ELA and visual active learning could be used together, but in this paper we focus only on characterizing the potential bias incurred by a non-visual approximate annotation method.

Attribute Annotation using the ELA

For ELA attribute annotation, we ask workers to label one attribute at a time. We cannot use the UI from Fig. 2b. Instead we ask the AMT workers to select all positive examples of a single attribute for a set of images from a given category, example shown in Fig. 2c. We elect to ask for fewer annotations per HIT in the ELA stage (50 object-attribute pairs) than in the exhaustive stage (200 object-attribute pairs). This choice was made to lessen worker fatigue and improve performance. The difference in worker performance for the exhaustive and ELA HITs is discussed more in the supplemental materials.

Thus far we have collected approximately 3.4M object-attribute pairs. Apart from the exhaustive labels we used to bootstrap ELA annotation, we have collected at least 20 attributes for 24,492 objects at a cost of **$2,469** ($\sim$ 24k objects $\times$ 20 attributes/50 attributes per HIT $\times$ 3 repeat workers $\times$ $0.084 per HIT). If we used the exhaustive annotation method, this would have cost **$8,817**.

In the end, the COCO Attributes dataset has a variety of popular and rare attributes. 75 % of attributes have more than 216 positive examples in the dataset, 50 % have more than 707, and 25 % have more than 2511. Figure 5 shows some qualitative examples from COCO Attributes.

6 Attribute Classification

Ultimately, attribute labels are only useful if visual classifiers can be built from them. To verify the detectability of our attributes, we trained independent clas-

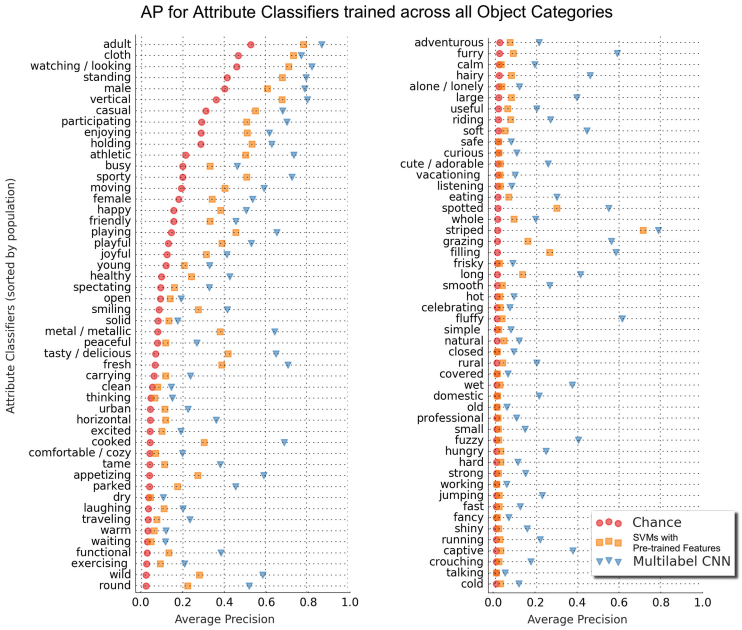


Fig. 6. *Average Precision vs. Chance.* Performance is shown for 100 randomly selected attributes. Attribtues are sorted in descending order by their population in the dataset. Each yellow square represents an SVM that was trained using pre-trained CNN features to recognize that particular attribute. Each blue triangle represents the AP for that attribute calculated on the full multi-label test set predictions of our fine-tuned CNN. (Color figure online)

sifiers for each attribute, agnostic of object category. Figure 6 shows AP scores of 100 randomly selected attribute classifiers.

To train the attribute classifiers, features for each object instance's bounding box were extracted using the pre-trained Caffe hybridCNN network released with the publication of the Places Dataset [28,29]. For the features used in these classifiers, we take the output of the final fully connected layer, randomly subsample 200 dimensions, and apply power normalization as per the recommendations of [30]. We then train a linear SVM using a set of object instances that have 20 or more attributes annotated with the ELA. Subsampling the FC7 activations to 200D actually leads to higher performance than using all activations (4096D), with an average increase of 0.012 AP across all attribute classifiers. Chance is calculated as the ratio of true positives to total training examples for each attribute.

As a counterpoint to recognizing attributes in isolation, we trained a multi-label CNN to simultaneously predict all attributes for a novel test image. We created this network by fine-tuning the BVLC reference network from the Caffe library [29]. Our attribute network uses a sigmoid cross-entropy loss layer instead of a softmax layer to optimize for multi-label classification, as suggested by

previous research in multi-label and attribute classification [31–33]. The fine-tuning was accomplished with SGD with momentum slightly higher than the reference net, but with learning rate lower and regularization stronger to account for the sparsity of positive labels in the training set. This network is trained with a the full, multi-label attribute vector for each object in the train and test sets. Unlabeled attributes are assumed to be negative after 20 rounds of ELA.

In Fig. 6 the objects in the training set for all classifiers shown are members of the COCO 'train2014' set, and test instances are members of the 'val2014' set. In order to compare the individual SVMs and the multilabel CNN, we used the same training and test sets for all classifiers. The train/test sets were composed of 50k object instances labeled with the ELA with 20 or more attributes. All unlabeled attributes were assumed false for both the SVMs and CNN. The train/test set split was 30k/20k and contained instances from COCO train and val sets respectively.

Figure 6 compares the per attribute AP over the test set predictions from our multi-label CNN to the independent SVMs trained independently for each attribute. This plot shows that exploiting the correlations between attributes often improves classifier performance for the CNN compared to the independent SVMs, especially for rarer attributes. Overall attributes the mean chance score is 0.08 AP, mean SVM performance is 0.18 AP, and mean CNN performance is 0.35 AP. This experiment shows the benefits of exploiting multilabel co-occurrence information with a 0.17 mAP over using pre-trained features.

7 Future Work

Work on COCO Attributes is ongoing. Workers are continuously submitting new ELA HITs for the remaining 'person', 'animal', 'vehicle', and 'food' instances from the COCO dataset. The set of objects could easily be expanded to comprise more of the COCO categories. More categories would necessitate more attributes, but our attribute discovery process combined with the ELA are capable of scaling up the annotation effort effectively. Further analysis of alternative selection methods could result in improved recall for low numbers of attribute queries. This economical annotation method begs to be used on larger dataset annotation efforts.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
2. Geman, D., Geman, S., Hallonquist, N., Younes, L.: Visual turing test for computer vision systems. Proc. Natl. Acad. Sci. **112**(12), 3618–3623 (2015)
3. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR (2009)
4. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009)

5. Rohrbach, M., Stark, M., Szarvas, G., Gurevych, I., Schiele, B.: What helps where- and why? semantic relatedness for knowledge transfer. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 910–917. IEEE (2010)
6. Deng, J., Krause, J., Fei-Fei, L.: Fine-grained crowdsourcing for fine-grained recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013
7. Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 438–451. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15561-1_32
8. Patterson, G., Hays, J.: Sun attribute database: discovering, annotating, and recognizing scene attributes. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2751–2758. IEEE (2012)
9. Dow, S., Kulkarni, A., Klemmer, S., Hartmann, B.: Shepherding the crowd yields better work. In: Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, pp. 1013–1022. ACM (2012)
10. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., et al.: Visual genome: connecting language and vision using crowdsourced dense image annotations. arXiv preprint arXiv:1602.07332 (2016)
11. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical report CNS-TR-2011-001, California Institute of Technology (2011)
12. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 663–676. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15549-9_48
13. Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: CVPR (2011)
14. Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: CVPR (2010)
15. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: ICCV (2009)
16. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
17. Bourdev, L., Maji, S., Malik, J.: Describing people: a poselet-based approach to attribute classification. In: 2011 International Conference on Computer Vision, pp. 1543–1550. IEEE (2011)
18. Vedaldi, A., Mahendran, S., Tsogkas, S., Maji, S., Girshick, R., Kannala, J., Rahtu, E., Kokkinos, I., Blaschko, M.B., Weiss, D., et al.: Understanding objects in detail with fine-grained attributes. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3622–3629. IEEE (2014)
19. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: image search with relative attribute feedback. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
20. Parikh, D., Grauman, K.: Relative attributes. In: ICCV (2011)
21. Vijayanarasimhan, S., Grauman, K.: Large-scale live active learning: training object detectors with crawled data and crowds. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1449–1456. IEEE (2011)
22. Abramson, Y., Freund, Y.: Active learning for visual object recognition. Technical report, UCSD (2004)

23. Collins, B., Deng, J., Li, K., Fei-Fei, L.: Towards scalable dataset construction: an active learning approach. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 86–98. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_8
24. Patterson, G., Van Horn, G., Belongie, S., Perona, P., Hays, J.: Tropel: crowdsourcing detectors with minimal training. In: Third AAAI Conference on Human Computation and Crowdsourcing (2015)
25. Deng, J., Russakovsky, O., Krause, J., Bernstein, M.S., Berg, A., Fei-Fei, L.: Scalable multi-label annotation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3099–3102. ACM (2014)
26. Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., Adam, H.: Large-scale object classification using label relation graphs. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 48–64. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_4
27. Sandhaus, E.: The new york times annotated corpus. Linguist. Data Consortium Philadelphia **6**(12), e26752 (2008)
28. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems, pp. 487–495 (2014)
29. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
30. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. arXiv preprint arXiv:1403.6382 (2014)
31. Shankar, S., Garg, V.K., Cipolla, R.: Deep-carving: discovering visual attributes by carving deep neural nets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3403–3412 (2015)
32. Gong, Y., Jia, Y., Leung, T., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. arXiv preprint arXiv:1312.4894 (2013)
33. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: TagProp: discriminative metric learning in nearest neighbor models for image auto-annotation. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 309–316. IEEE (2009)