# Collecting a Large-Scale Dataset of Fine-Grained Cars

Jonathan Krause[1], Jia Deng[1], Michael Stark[1,2], and Li Fei-Fei[1]

[1]Computer Science Department, Stanford University
[2]Max Planck Institute for Informatics

## 1. Introduction

In this work we introduce a large-scale, fine-grained dataset of cars. This dataset, consisting of 197 classes and 16,185 images, represents an order of magnitude increase in size over the only existing fine-grained car dataset [7] (14 classes, 1,904 images) and is comparable in size to the largest fine-grained datasets publicly available [9, 3]. The goals of this work are twofold: 1) to describe the difficulties encountered when collecting such a dataset and 2) to present baseline performance for two state-of-the-art methods.

## 2. Dataset Collection

**Finding Distinctive Classes.** The first challenging step in collecting a dataset where the categories are man-made, rather than naturally defined, is to determine a list of distinctive classes. Unlike naturally-occurring objects [9, 3, 4, 6], the class list of cars changes on a yearly basis, the appearance of some models of cars remains constant from year to year, and typical car websites may even list cars which differ only in terms of non-visual features, *e.g.* their engine, as separate classes. We initialize our class list by crawling a popular car website for a list of all types of cars made since 1990. Then we apply an aggressive deduplication procedure based on perceptual hashing [12] to a limited number of provided example images for these classes. Perceptual hashing maps each image into a binary vector, where image similarity is determined by Hamming distance, and deduplication consists of merging classes where the minimal distance between their sets of example images is less than some threshold $\tau$. From these merged categories, we subsample 197 categories for further annotation.

**Image Collection.** Candidate images for each class were collected from Flickr, Google, and Bing. To reduce annotation cost and ensure diversity in the data, the candidate images for each class were deduplicated using the same perceptual hash algorithm [12], yielding several thousand candidate images for each target classes. These images were then put on Amazon Mechanical Turk (AMT) in order to verify whether they belong to their respective target classes.

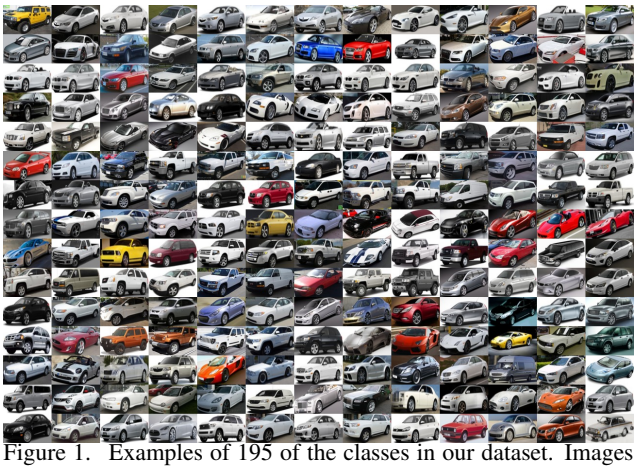

Figure 1. Examples of 195 of the classes in our dataset. Images have been chosen to share a common viewpoint.

**Training Annotators.** The primary challenge in crowdsourcing the collection of a fine-grained dataset is that workers know neither of the existence of many of the fine-grained car categories nor how to differentiate them. To compensate, we first provided specific instructions about how to approach fine-grained categorization, including examples of two fine-grained classes along with highlighted common and discriminating parts. Then each potential annotator must go through a qualification task (a set of particularly hard examples of the actual annotation task), and is not allowed to work on the full task unless he or she obtains a sufficiently high accuracy. In the real annotation task, we also provide a set of positive and negative example images for the car class a worker is annotating, drawing the negative examples from classes known a priori to be similar to the target class based on metadata such as the make, model, and year of each class. This allows the workers the better discriminate the target class from other classes where mistakes may otherwise be commonly made.

**Modeling Annotator Quality.** Even after training, workers differ in quality by large margins. Some workers may legitimately be car experts doing the tasks for fun, while others are spammers who only put forth enough effort to pass the qualification task. To tackle this problem, we use the Get Another Label (GAL) system developed by Ipeirotis *et al*. [2], which simultaneously estimates the probability a candidate image belongs to its target class and determines a quality level for each worker via an EM-like procedure. These estimates are determined from two sources of information: 1) mutual agreement of workers on image labels and 2) "gold standard" images placed throughout an annotation task and for which we know the correct label. In our case, gold standard images are flipped versions of example images we have for each class, including both positive and negative examples. Candidate images whose probability of belonging to the target class exceeds a specified threshold are then added to the set of images for that category. An additional benefit of this approach is that we can use estimated worker qualities produced by the GAL system to make our annotation task more efficient by making the number of candidate images we show to each worker dependent on our quality estimate of the worker, showing more images to bad workers and fewer to workers whose quality estimate is high. This encourages good workers to do our task while driving poor workers away.

After obtaining images for each of the 197 target classes, we collect a bounding box for each image via AMT, using a quality-controlled system provided to us by the authors of [8]. Finally, an additional stage of deduplication is performed on the images when cropped to their bounding boxes. This is necessary because it is often is the case that one image is simply a zoomed-out version of another image or that the full images differ by some other transformation which does not affect the appearance of the car itself. Fig. 1 shows example dataset images, selected to all be at a common viewpoint, and Fig. 2 gives a distribution of the higher-level categories within our dataset.

## 3. Experimental Procedure

For evaluation, we separate the dataset into two splits: 50% training and 50% testing. Images are cropped to their ground truth bounding box, as is standard in fine-grained classification. For baselines, we use Locality-Constrained Linear Coding (LLC) [10] and a randomized version of BubbleBank (BB) [1], which is similar to the randomized technique of [11]. For LLC, a codebook of size 4,096 and three levels of SPM [5] are used. For BB, 10k random bubbles are used. LLC achieved an accuracy of 69.5%, and the randomized BB had an accuracy of 63.6%. This relatively high performance suggests that fine-grained car classification 1) is a promising area for application in the near future and 2) can be pushed to an even larger scale.

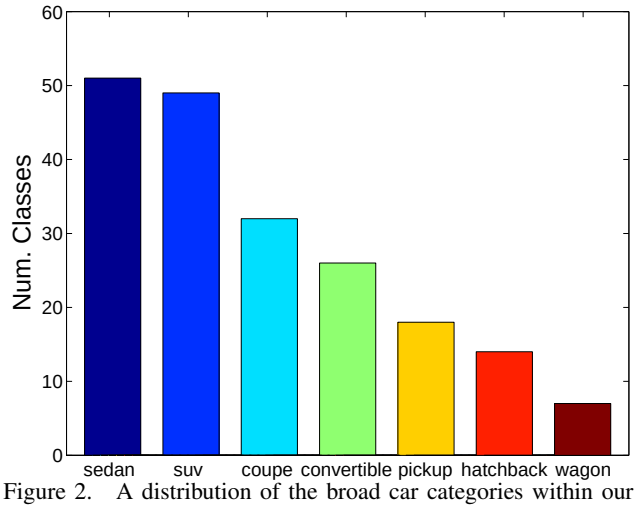

Figure 2. A distribution of the broad car categories within our dataset.

## References


[1] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 2

[2] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *WS ACM SIGKDD*, 2010. 2

[3] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR-WFGVC*, 2011. 1

[4] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*. 2012. 1

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2

[6] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006. 1

[7] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012. 1

[8] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI-WS*, 2012. 2

[9] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1

[10] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR'10*. 2

[11] B. Yao, G. Bradski, and L. Fei-Fei. A codebook- and annotation-free approach for fine-grained image categorization. In *CVPR'12*. 2

[12] C. Zauner. Implementation and benchmarking of perceptual image hash functions. *Master's thesis, Austria*. 1