Caltech-256 Object Category Dataset

Greg Griffin, Alex Holub and Pietro Perona

Abstract

We introduce a challenging set of 256 object categories containing a total of 30607 images. The original Caltech-101 [1] was collected by choosing a set of object categories, downloading examples from Google Images and then manually screening out all images that did not fit the category. Caltech-256 is collected in a similar manner with several improvement: a) the number of categories is more than doubled, b) the minimum number of images in any category is increased from 31 to 80, c) artifacts due to image rotation are avoided and d) a new and larger clutter category is introduced for testing background rejection. We suggest several testing paradigms to measure classification performance, then benchmark the dataset using two simple metrics as well as a state-of-the-art spatial pyramid matching [2] algorithm. Finally we use the clutter category to train an interest detector which rejects uninformative background regions.

1 Introduction

Recent years have seen an explosion of work in the area of object recognition [1, 2, 3, 4, 5, 6]. Several datasets have emerged as standards for the community, including the Coil [7], MIT-CSAIL [8] PASCAL VOC [9], Caltech-6 and Caltech-101 [1] and Graz [10] datasets. These datasets have become progressively more challenging as existing algorithms consistently saturated performance. The Coil set contains objects placed on a black background with no clutter. The Caltech-6[1] consists of 3738 images of cars, motorcycles, airplanes, faces and leaves. The Caltech-101[2] is similar in spirit to the Caltech-6 but has many more object categories, as well as hand-clicked silhouettes of each object. The MIT-CSAIL database contains more than 77,000 objects labeled within 23,000 images that are shown in a variety of environments. The number of labeled objects, object categories and region categories increases over time thanks to a publicly available LabelMe [11] annotation tool. The PASCAL VOC 2006 database contains 5,304 images where 10 categories are fully annotated. Finally, the Graz set contains three object categories in difficult viewing conditions. These and other standardized sets of categories allow users to compare the performance of their algorithms in a consistent manner.

Here we introduce the Caltech-256[3]. Each category has a minimum of 80 images (compared to the Caltech-101 where some classes have as few as 31

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech6

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech256

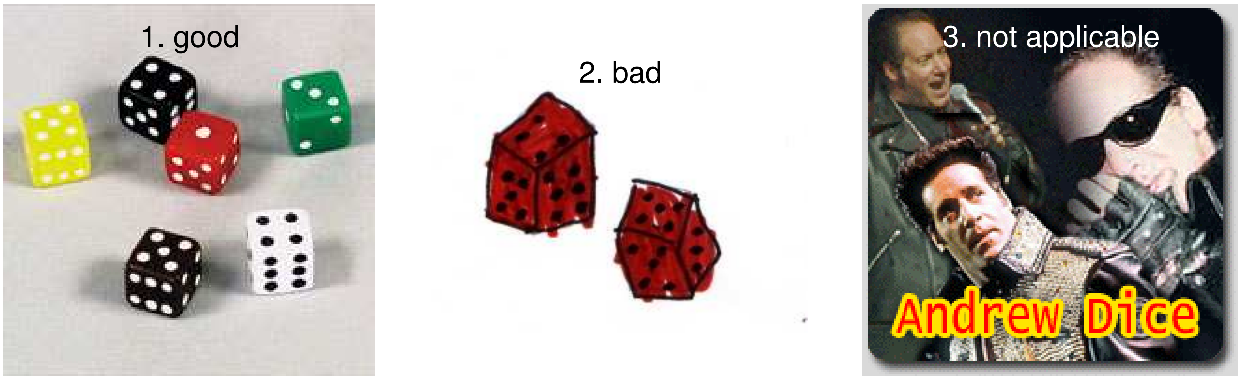

Figure 1: Examples of a 1, 2 and 3 rating for images downloaded using the keyword *dice.*

images). In addition we do not left-right align the object categories as was done with the Caltech-101, resulting in a more formidable set of categories.

Because Caltech-256 images are harvested from two popular online image databases, they represent a diverse set of lighting conditions, poses, backgrounds, image sizes and camera systematics. The categories were hand-picked by the authors to represent a wide variety of natural and artificial objects in various settings. The organization is simple and the images are ready to use, without the need for cropping or other processing. In most cases the object of interest is prominent with a small or medium degree of background clutter.

Dataset	*Released*	*Categories*	*Images*	*Images Per Category*			
			Total	*Min*	*Med*	*Mean*	*Max*
Caltech-101	2003	102	9144	31	59	90	800
Caltech-256	2006	257	30607	80	100	119	827

Figure 2: Summary of Caltech image datasets. There are actually 102 and 257 categories if the *clutter* categories in each set are included.

In Section 2 we describe the collection procedures for the dataset. In Section 3 we give paradigms for testing recognition algorithms, including the use of the background *clutter* class. Example experiments are provided in Section 4. Finally in Section 5 we conclude with a general discussion of advantages and disadvantages of the set.

2 Collection Procedure

The object categories were assembled in a similar manner to the Caltech-101. A small group of vision dataset users were asked to supply the names of roughly 300 object categories. Images from each category were downloaded from both Google[4] and PicSearch[5] using scripts[6]. We required that the minimum size in either aspect be 100 with no upper range. Typically this procedure resulted in about $400 - 600$ images from each category. Duplicates were removed by detecting images which contained over 15 similar SIFT descriptors [12].

[4] `http://images.google.com`
[5] `http://www.picsearch.com`
[6] Based on software written by Rob Fergus

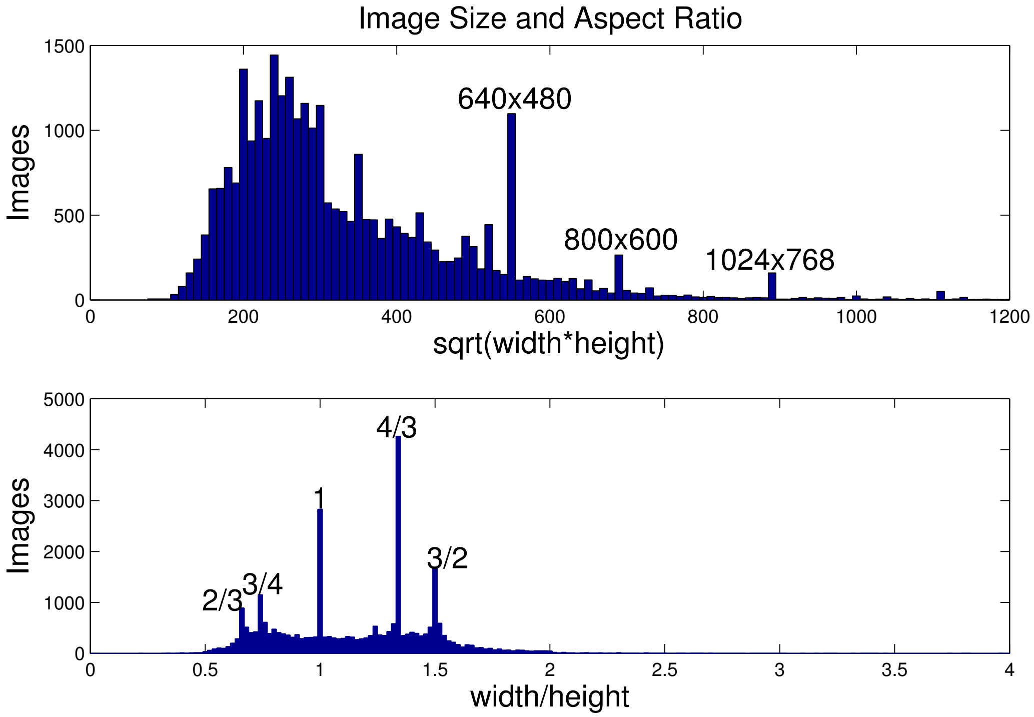

Figure 3: Distribution of image sizes as measured by $\sqrt{\mathrm{width} \cdot \mathrm{height}}$, and aspect ratios as measured by width/height. Some common image sizes and aspect ratios that are overrepresented are labeled above the histograms. Overall in Caltech-256 the mean image size is 351 pixels while the mean aspect ratio is 1.17.

The images obtained were of varying quality. We asked 4 different subjects to rate these images using the following criteria:

1. *Good*: A clear example of the visual category
2. *Bad*: A confusing, occluded, cluttered or artistic example
3. *Not Applicable*: Not an example of the object category

Sorters were instructed to label the image *bad* if either: (1) the image was very cluttered, (2) the image was a line drawing, (3) the image was an abstract artistic representation, or (4) the object within the image occupied only a small fraction of the image. If the image contained no examples of the visual category it was labeled *not applicable.* Examples of each of the 3 ratings are shown in Figure 1.

The final set of images included in Caltech-256 are the ones that passed our size and duplicate checks and were also rated *good.* Out of 304 original categories 48 had less than 80 *good* images and were dropped, leaving 256 categories. Figure 3 shows the distribution of the sizes of these final images.

In Caltech-101, categories such as *minaret* had a large number of images that were artificially rotated, resulting in large black borders around the image. This rotation created artifacts which certain recognition systems exploited resulting in deceptively high performance. This made such categories artificially easy to identify. We have not introduced such artifacts into this set and collecting an entirely new *minaret* category which was not artificially rotated.

In addition we did not consistently right-left align the object categories as was done in Caltech-101. For example *airplanes* may be facing in either the left or right direction now. This gives a better idea of what categorization performance would be like under realistic conditions, unlike that Caltech-101 *airplanes* which are all facing right.

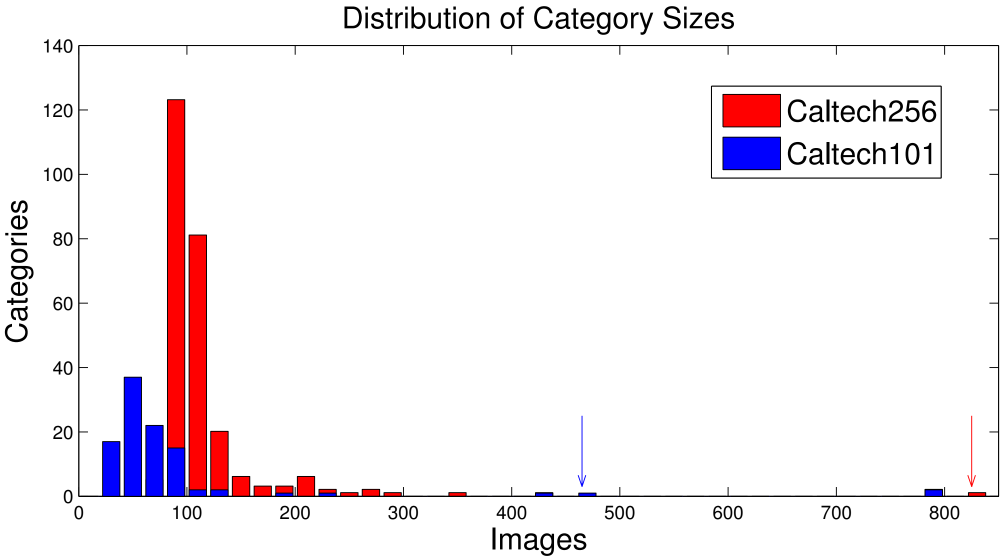

Figure 4: Histogram showing number of images per category. Caltech-101's largest categories *faces-easy* (435), *motorbikes* (798), *airplanes* (800) are shared with Caltech 256. An additional large category *t-shirt* (358) has been added. The *clutter* categories for Caltech-101 (467) and 256 (827) are identified with arrows. This figure should be viewed in color.

2.1 Image Relevance

We compiled statistics on the downloaded images to examine the typical yield of *good* images. Figure 5 summarizes the results for images returned by Google. As expected, the relevance of the images decreases as more images are returned. Some categories return more pertinent results than others. In particular, certain categories contain dual semantic meanings. For example the category *pawn* yields both the chess piece and also images of pawn shops. The category *egg* is too ambiguous, because it yields images of whole eggs, egg yolks, Faberge Eggs, etc. which are not in the same visual category. These ambiguities were often removed with a more specific keyword search, such as *fried-egg*.

When using Google images alone, 25.6% of the images downloaded were found to be *good*. To increase the precision of image downloading we augmented the Google search with PicSearch.

Since both search engines return largely non-overlapping sets of images, the overall precision for the initial set of downloaded images increased, as both returned a high fraction of good images initially. Now 44.4% of the images were usable. The true overall precision was slightly lower as there was some overlap between the Google and PicSearch images. A total of 9104 *good* images were gathered from PicSearch and 20677 from Google, out of a total of 92652 downloaded images. Thus the overall sorting efficiency was 32.1%.

2.2 Categories

The category numbering provides some insight into which categories are similar to an existing category. Categories $\mathcal{C}_1 ... \mathcal{C}_{250}$ are relatively independent of one another, whereas categories $\mathcal{C}_{251} ... \mathcal{C}_{256}$ are closely related to other categories. These are *airplane-101*, *car-side-101*, *faces-easy-101*, *greyhound*, *tennis-shoe* and *toad*, which are closely related to *fighter-jet*, *car-tire*, *people*, *dog*, *sneaker* and *frog* respectively. We felt these 6 category pairs would be the most likely to be confounded with one another, so it would be best to remove one of each pair

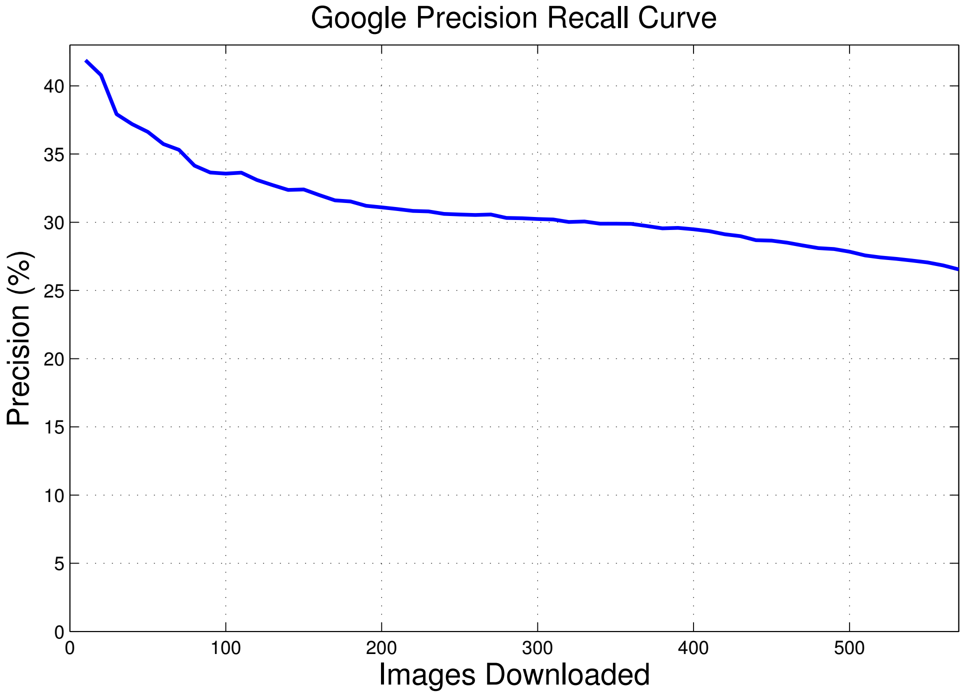

Figure 5: Precision of images returned by Google. This is defined as the total number of images rated *good* divided by the total number of images downloaded (averaged over many categories). As more images are download, it becomes progressively more difficult to gather large numbers of images per object category. For example, to gather 40 good images per category it is necessary to collect 120 images and discard 2/3 of them. To gather 160 good images, expect to collect about 640 images and discard 3/4 of them.

from the confusion matrix, at least for the standard benchmarking procedure[7].

2.3 Taxonomy

Figure 6 shows a taxonomy of the final categories, grouped by animate and inanimate and other finer distinctions. This taxonomy was compiled by the authors and is somewhat arbitrary; other equally valid hierarchies can be constructed. The largest 30 categories from Caltech-101 (shown in green) were included in Caltech-256, with additional images added as needed to boost the number of images in each category to at least 80. Animate objects - 69 categories in all - tend to be more cluttered than the inanimate objects, and harder to identify. A total of 12 categories are marked in red to denote a possible relation with some other visual category.

2.4 Background

Category $\mathcal{C}_{257}$ is *clutter*[8]. For several reasons (see subsection 3.4) it is useful to have such a background category, but the exact nature of this category will vary from set to set. Different backgrounds may be appropriate for different

[7] While *horseshoe-crab* may seem to be a specific case of *crab*, the images themselves involve two entirely different sub-phylum of Arthropoda, which have clear differences in morphology. We find these easy to tell apart whereas *frog* and *toad* differences can be more subtle (none of our sorters were herpetologists). Likewise we feel that *knife* and *swiss-army-knife* are not confounding, even though they share some characteristics such as blades.

[8] For purposes here we will use the terms *background* and *clutter* interchangeably to indicate the absence or near-absence of any objects categories

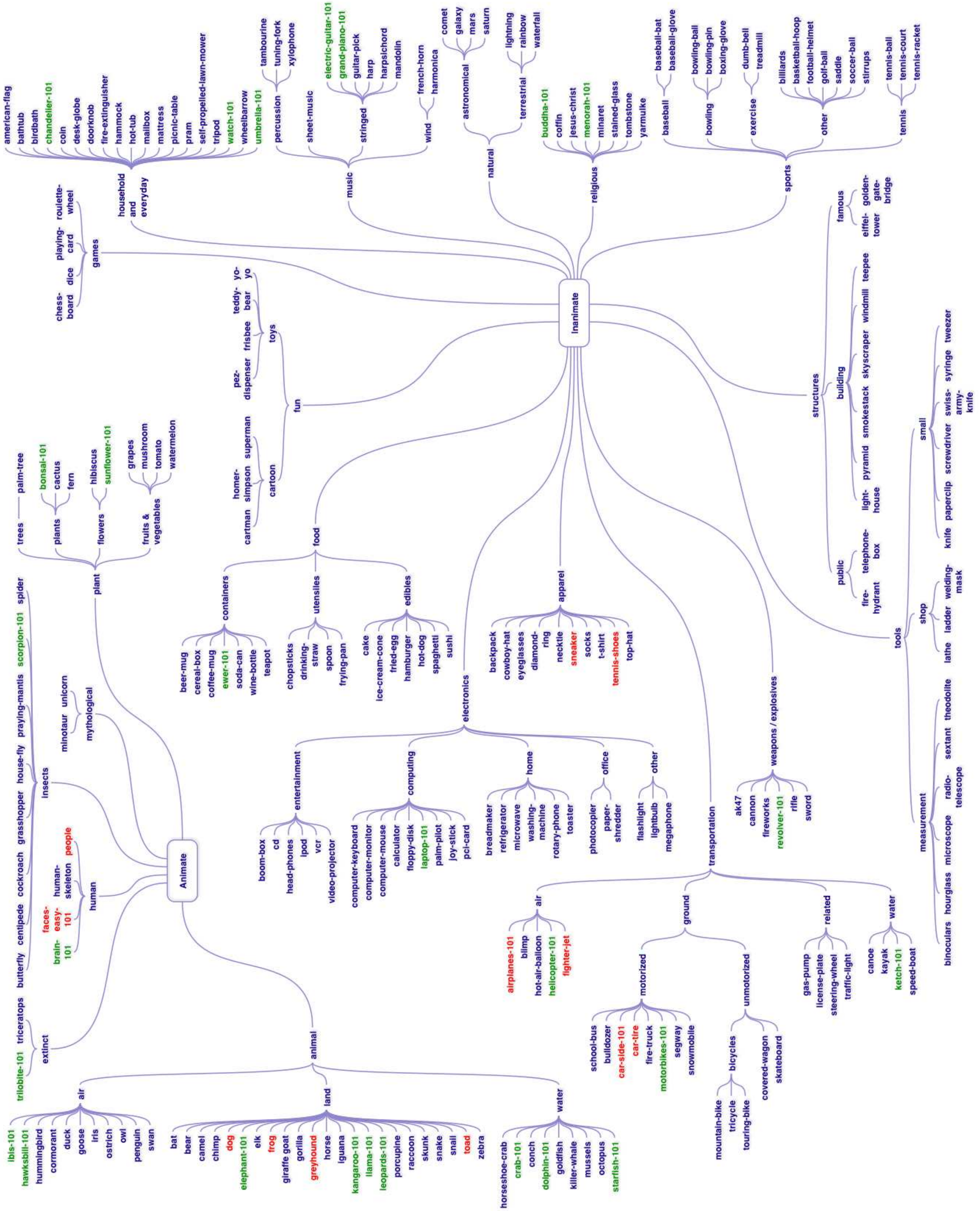

Figure 6: A taxonomy of Caltech-256 categories created by hand. At the top level these are divided into animate and inanimate objects. Green categories contain images that were borrowed from Caltech-101. A category is colored red if it overlaps with some other category (such as *dog* and *greyhound*).

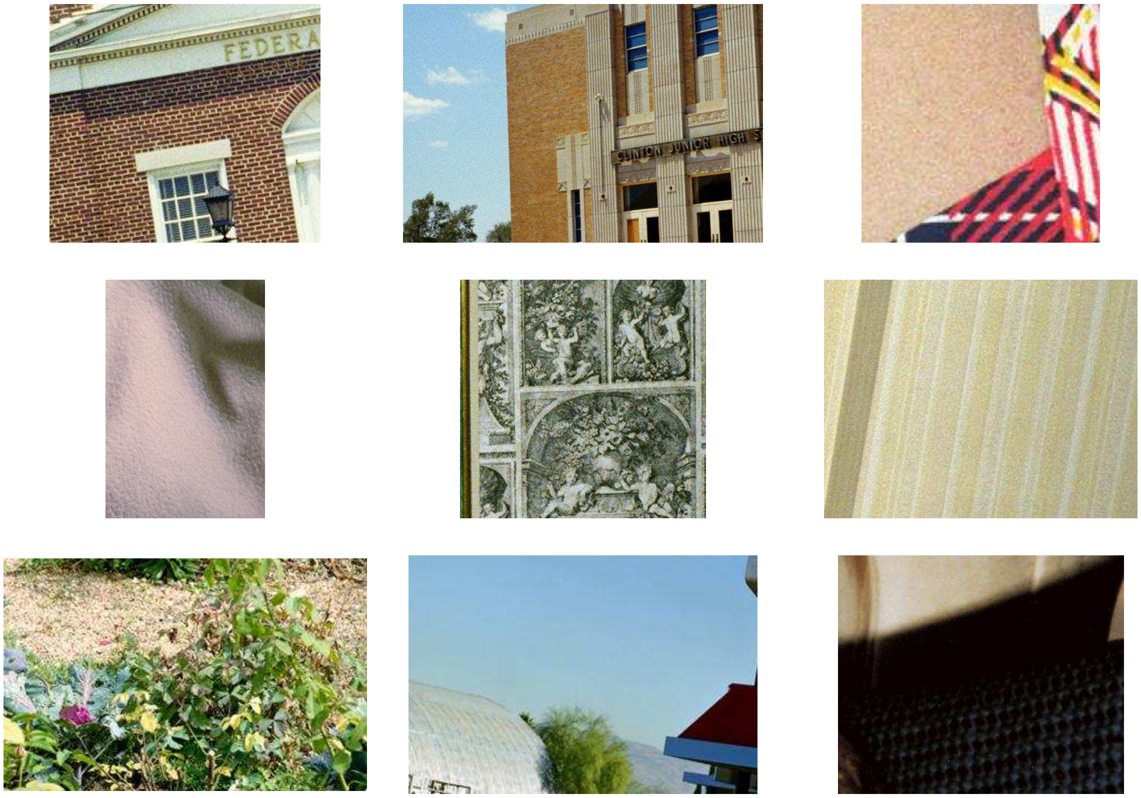

Figure 7: Examples of *clutter* generated by cropping the photographs of Stephen Shore [13, 14].

applications, and the statistics of a given background category can effect the performance of the classifier [15].

For instance Caltech-6 contains a background set which consists of random pictures taken around Caltech. The image statistics are no doubt biased by their specific choice of location. The Caltech-101 contains a set of background images obtained by typing the keyword “things” into Google. This can turn up a wide variety of objects not in Caltech-101. However these images may or may not contain objects of interest that the user would wish to classify.

Here we choose a different approach. The *clutter* category in Caltech-256 is derived by cropping 947 images from the pictures of photographer Stephen Shore [13, 14]. Images were cropped such that the final image sizes in the clutter category are representative of the distribution of images sizes found in all the other categories (figure 3). Those cropped images which contained Caltech-256 categories (such as people and cars) were manually removed, with a total of 827 *clutter* images remaining. Examples are shown in Figure 7.

We feel that this is an improvement over our previous clutter categories, since the images contain clutter in a variety of indoor and outdoor scenes. However it is still far from perfect. For example some visual categories such as grass, brick and clouds appear to be over-represented.

3 Benchmarks

Previous datasets suffered from non-standard testing and training paradigms, making direct comparisons of certain algorithms difficult. For instance, results reported by Grauman [16] and Berg [17] were not directly comparable as Berg

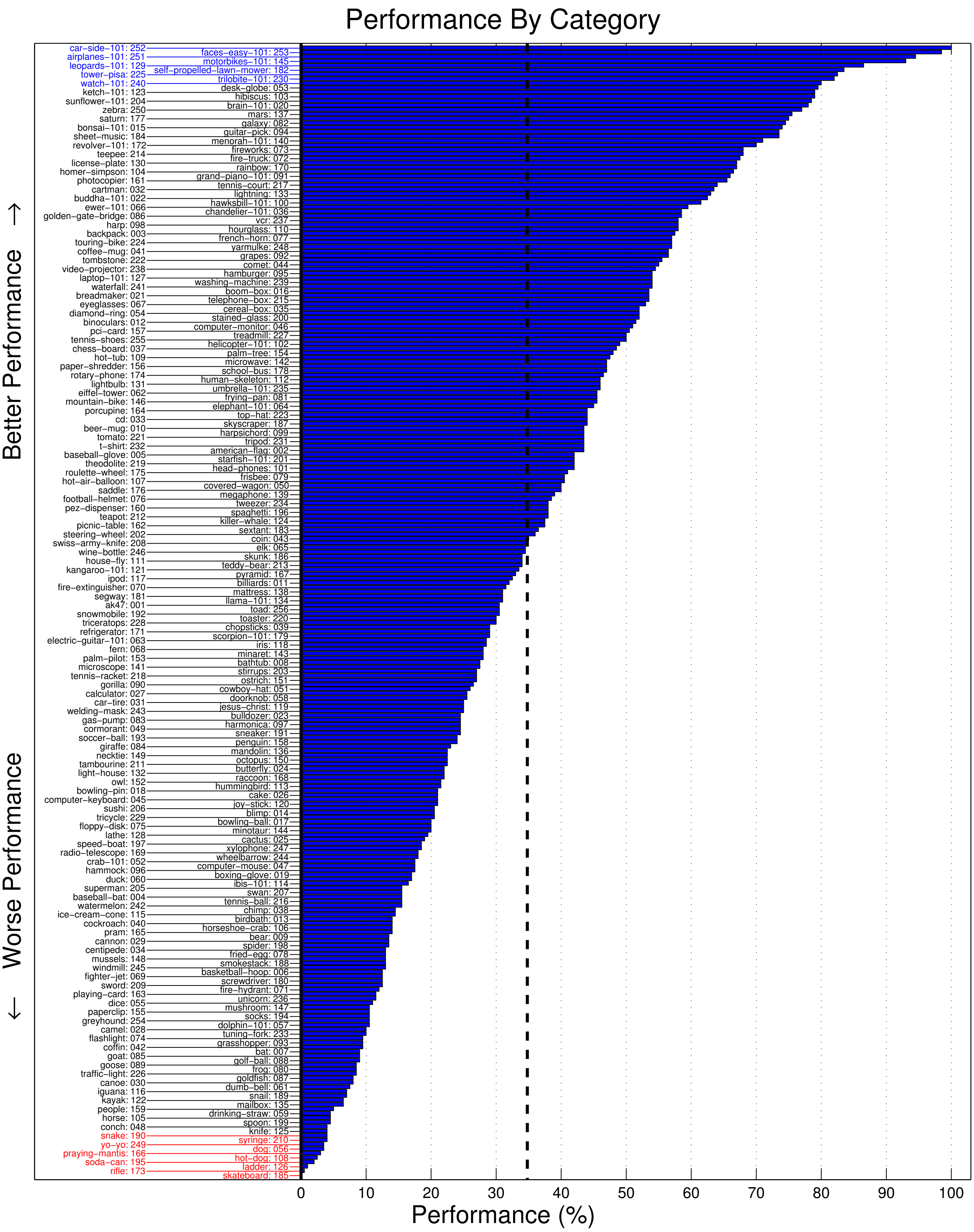

Figure 8: Performance of all 256 object categories using a typical pyramid match kernel [2] in a multi-class setting with $N_{\mathrm{train}} = 30$. This performance corresponds to the diagonal entries of the confusion matrix, here sorted from largest to smallest. The ten best performing categories are shown in blue at the top left. The ten worst performing categories are shown in red at the bottom left. Vertical dashed lines indicate the mean performance.

used only 15 training while Grauman used 30 training examples [9]. Some authors used the same number of test examples for each category, while other did not. This can be confusing if the results are not normalized in a consistent way. For consistent comparisons between different classification algorithms, it is useful to adopt standardized training and testing procedures

3.1 Performance

First we select N_{train} and N_{test} images from each class to train and test the classifier. Specifically $N_{\text{train}} = 5, 10, 15, 20, 25, 30, 40$ and $N_{\text{test}} = 25$.

Each test image is assigned to a particular class by the classifier. Performance of each class $\mathcal{C}$ can be measured by determining the fraction of test examples for class $\mathcal{C}$ which are correctly classified as belonging to class $\mathcal{C}$. The cumulative performance is calculated by counting the total number of correctly classified test images N_{test} within each of N_{class} classes. It is of course important to weight each class equally in this metric. The easiest way to guarantee this is to use the same number of test images for each class. Finally, better statistics are obtained by averaging the above procedure multiple times (ideally at least 10 times) to reduce uncertainty.

The exactly value of N_{test} is not important. For Caltech-101 values higher than $N_{\text{train}} = 30$ are impossible since some categories contain only 31 images. However Caltech-256 has at least 80 images in all categories. Even a training set size of $N_{\text{train}} = 75$ leaves $N_{\text{test}} \geq 5$ available for testing in all categories.

The confusion matrix $\mathcal{M}_{ij}$ illustrates classification performance. It is a table where each element i, j stores the fraction of the test images from category $\mathcal{C}_i$ that were classified as belonging to $\mathcal{C}_j$. Note that perfect classification would result in a table with ones along the main diagonal. Even if such a classification method existed, this ideal performance would not be reached for several reasons. Images in most categories contain instances of other categories, which is a built-in source of confusion. Also our sorting procedure is never prefect; there are bound to be some small fraction of incorrectly classified images in a dataset of this size.

Since the last 6 categories are redundant with existing categories, and *clutter* indicates the absence of any category, one might argue that only categories $\mathcal{C}_1 ... \mathcal{C}_{250}$ are appropriate for generating performance benchmarks. Another justification for removing these last 6 categories when measuring overall performance is that they are among the easiest to identify. Thus removing them makes the detection task more challenging[10].

However for better clarity and consistency, we suggest that authors remove only the *clutter* category, *generate a 256x256 confusion matrix* with the remaining categories, and report their performance results directly from the diagonal of this matrix[11]. Is also useful for authors to post the confusion matrix itself - not just the mean of the diagonal.

[9] It should be noted that Grauman achieved results surpassing those of Berg in experiments conducted later.

[10] As shown in figure 13, categories $\mathcal{C}_{251}$, $\mathcal{C}_{252}$ and $\mathcal{C}_{253}$ each yield performance above 90%

[11] The difference in performance between the 250x250 and 256x256 matrix is typically less than a percent

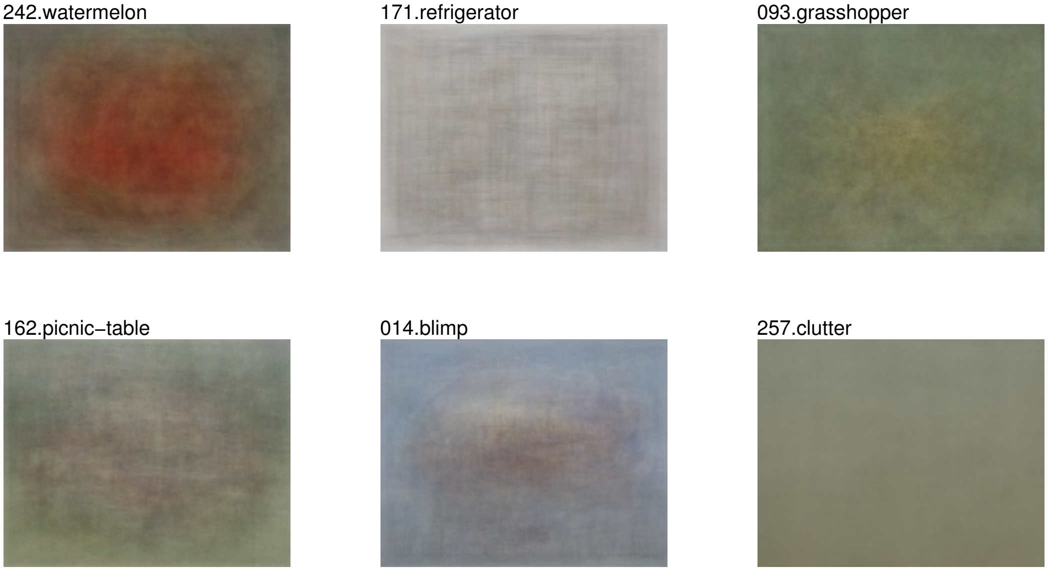

Figure 9: The mean of all images in five randomly chosen categories, as compared to the mean *clutter* image. Four categories show some degree of concentration towards the center while *refrigerator* and *clutter* do not.

3.2 Localization and Segmentation

Both Caltech-101 and the Caltech-256 contain categories in which the object may tend to be centered (Figure 9). Thus, neither set is appropriate for localization experiments, in which the algorithm must not only identify what object is present in the image but also where the object is.

Furthermore we have not manually annotated the images in Caltech-256 so there is presently no ground truth for testing segmentation algorithms.

3.3 Generality

Why not remove the last 6 categories from the dataset altogether? Closely related categories can provide useful information that is not captured by the standard performance metric. Is a certain *greyhound* classifier also good at identifying *dog*, or does it only detect specific breeds? Does a *sneaker* detector also detect images from *tennis-shoe*, a word which means essentially the same thing? If it does not, one might worry that the algorithm is over-training on specific features of the dataset which do not generalize to visual categories in the real world.

For this reason we plot rows 251..256 of the confusion matrix along with the categories which are most similar to these, and discuss the results in section 3.3.

3.4 Background

Consider the example of a Mars rover that moves around in its environment while taking pictures. Raw performance only tells us the accuracy with which objects are identified. Just as important is the ability to identify where there is an object of interest and where there is only uninteresting background. The rover cannot begin to understand its environment if background is constantly misidentified as an object.

The rover example also illustrates how the meaning of the word *background* is strongly dependent on the environment and the application. Our choice of background images for Caltech-256, as described in 2.4, is meant to reflect a variety of common (terrestrial) environments.

Here we generate an ROC curve that tests the ability of the classification algorithm to identify regions of interest. An ROC curve shows the ratio of false positives to true positives. In single-category detection the meaning of true positive and false positive is unambiguous. Imagine that a search window of varied size scans across an image employing some sort of bird classifier. Each true positive marks a successful detection of a bird inside the scan window while each false positive indicates an erroneous detection.

What do positive and negative mean in the context of multi-class classification? Consider a two-step process in which each search window is evaluated by a cascade [18] of two classifiers. The first classifier is an *interest* detector that decides whether a given window contains a object category or background. Background regions are discarded to save time, while all other images are passed to the second classifier. This more expensive multi-class classifier now attempts to identify which of the remaining 256 object categories best matches the region as described in 3.1.

Our ROC curve measures the performance of several *interest* classifiers. A false positive is any *clutter* image which is misclassified as containing an object of interest. Likewise true positive refers to an object of interest that is correctly identified. Here "object of interest" means any classification besides *clutter*.

4 Results

In this section we describe two simple classification algorithms as well as the more sophisticated spatial pyramid matching algorithm of Lazebnik, Schmid and Ponce [2]. Performance, generality and background rejection benchmarks are presented as examples for discussion.

4.1 Size Classifier

Our first classifier used only the width and height of each image as features. During the training phase, the width and height of all $256 \cdot N_{\text{train}}$ images are stored in a 2-dimensional space. Each test image is classified in a KNN fashion by voting among the 10 nearest neighbors to each image. The 1-norm Manhattan distance yields slightly better performance than the 2-norm Euclidean distance. As shown in Figure 12, this algorithm identifies the correct category for an image $3.7 \pm 0.6\%$ of the time when $N_{\text{train}} = 30$.

Although identifying the correct object category 3.7% of the time seems like paltry performance, we note that baseline (random guessing) would result in a performance of less than .25%. This illustrates a danger inherent in many recognition datasets: the algorithm can learn on ancillary features of the dataset instead of features intrinsic to the object categories. Such an algorithm will fail to identify categories if the images come from another dataset with different statistics.

As shown in

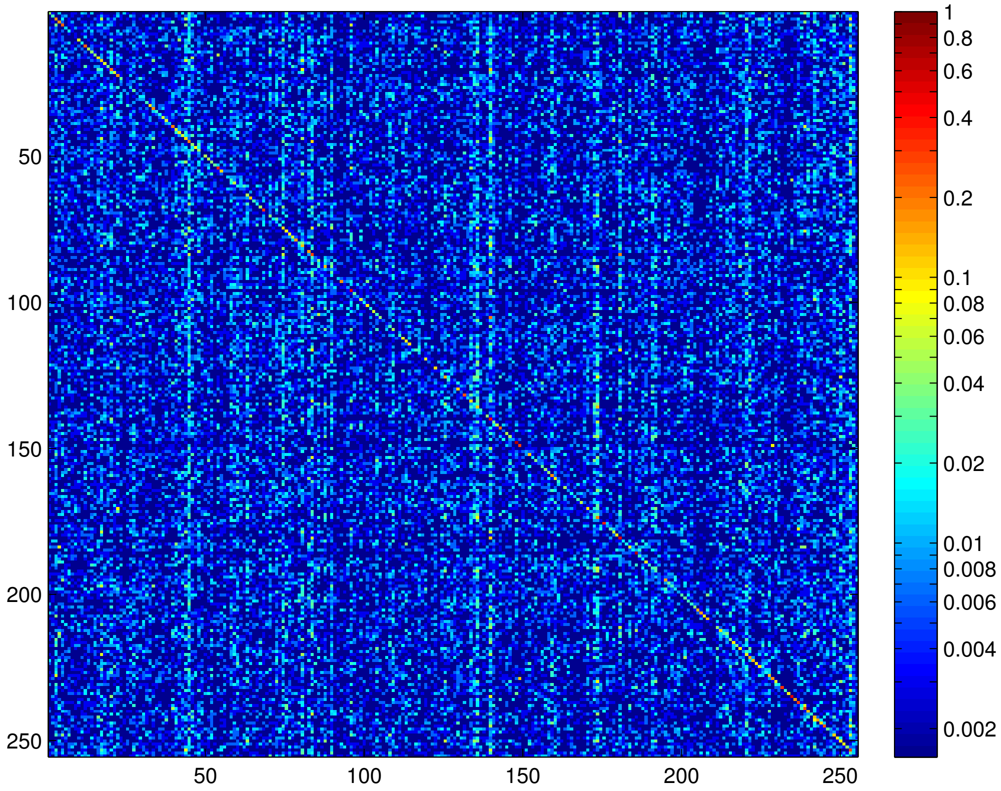

Figure 10: The 256×256 matrix $\mathcal{M}$ for the correlation classifier described in subsection 4.2. This is the mean of 10 separate confusion matrices generated for $N_{\text{train}} = 30$. A log scale is used to make it easier to see off-diagonal elements. For clarity we isolate the diagonal and row 82 *galaxy* and describe their meaning in Figure 11.

4.2 Correlation Classifier

The next classifier we employed was a correlation based classifier. All images were resized to $N_{dim} \times N_{dim}$, desaturated and normalized to have unit variance. The nearest neighbor was computed in the ${N_{dim}}^2$-dimensional space of pixel intensities. This is equivalent to finding the training image that correlates best with the test image, since

$$< (X - Y)^2 >=< X^2 > + < Y^2 > -2 < XY >= -2 < XY >$$

for images X,Y with unit variance. Again we use the 1-norm instead of the 2-norm because it is faster to compute and yields better classification performance.

Performance of $7.6 \pm 0.7\%$ at $N_{\text{train}} = 30$ is computed by taking the mean of the diagonal of the confusion matrix in Figure 10.

4.3 Spatial Pyramid Matching

As a final test we re-implement the spatial pyramid matching algorithm of Lazebnik, Schmid and Ponce [2] as faithfully as possible. In this procedure an SVM kernel is generating from matching scores between a set of training images. Their published Caltech-101 performance at $N_{\text{train}} = 30$ was $64.6 \pm 0.8\%$. Our own performance is practically the same.

As shown in Figure 12, performance on Caltech-256 is roughly half the performance achieved on Caltech-101. For example at $N_{\text{train}} = 30$ our Caltech-256 and Caltech-101 performance are $67.6 \pm 1.4\%$ and $34.1 \pm 0.2\%$ respectively.

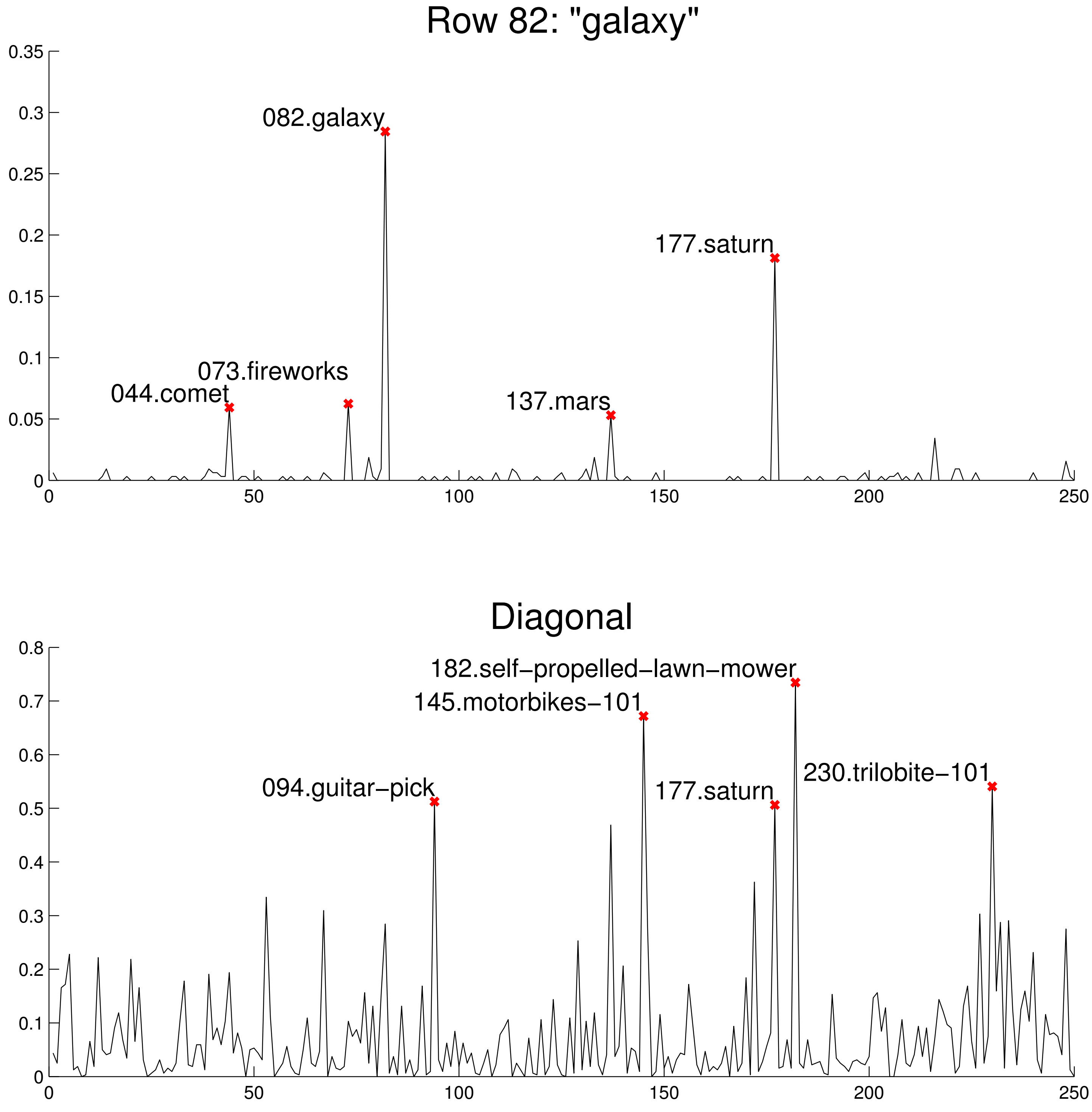

Figure 11: A more detailed look at the confusion matrix $\mathcal{M}$ from figure 10. Top: row 82 shows which categories were most likely to be confused with *galaxy*. These are: *galaxy*, *saturn*, *fireworks*, *comet* and *mars* (in order of greatest to least confusion). Bottom: the largest diagonal elements represent the categories that are easiest to classify with the correlation algorithm. These are: *self-propelled-lawn-mower*, *motorbikes-101*, *trilobite-101*, *guitar-pick* and *saturn*. All of these categories tend to have objects that are located consistently between images.

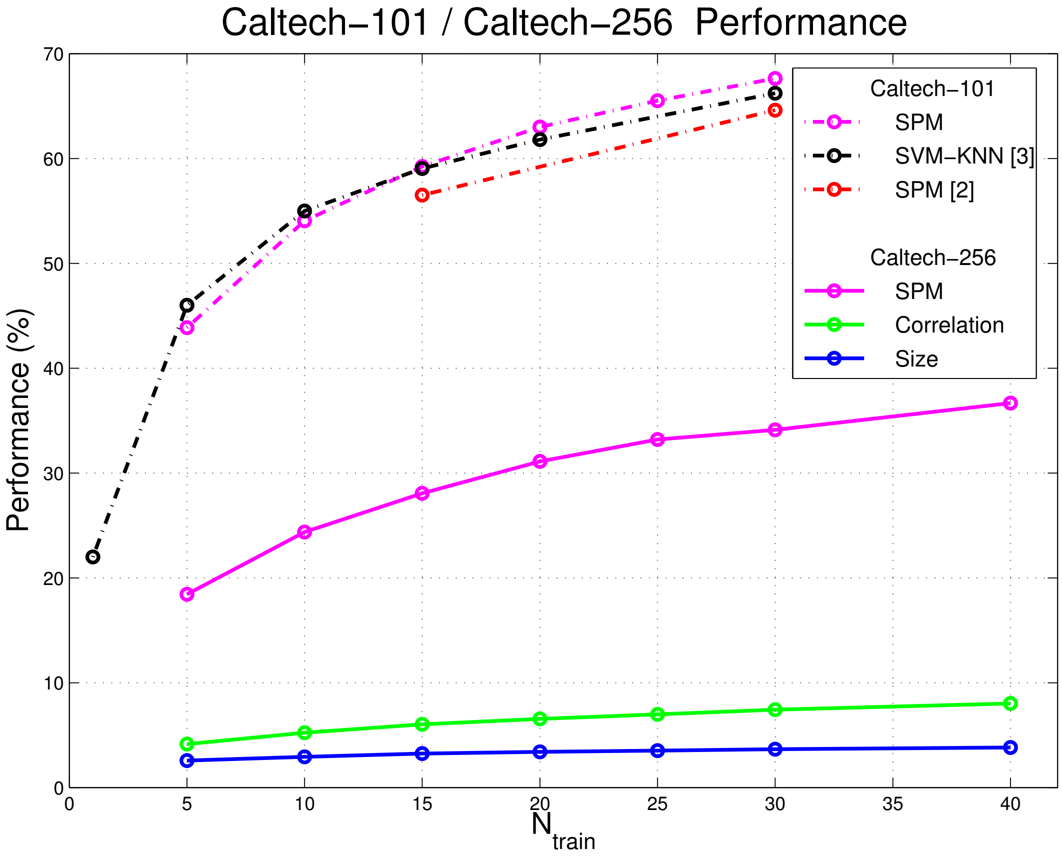

Figure 12: Performance as a function of N_{train} for Caltech-101 and Caltech-256 using the 3 algorithms discussed in the text. The spatial pyramid matching algorithm is that of Lazebnik, Schmid and Ponce [2]. We compare our own implementation with their published results, as well as the SVM-KNN approach of Zhang, Berg, Maire and Malik [3].

4.4 Generality

Figure 13 shows the confusion between six categories and their six confounding categories. We define the *generality* as the mean of the off-quadrant diagonals divided by the mean of the main diagonal. In this case, for $N_{\mathrm{train}} = 30$, the generality is $g = 0.145$.

What does g signify? Consider two extreme cases. If $g = 0.0$ then their is absolutely no confusion between any of the similar categories, including *tennis-shoe* and *sneaker*. This would be suspicious since it means the categorization algorithm is splitting hairs, ie. finding significant differences where none should exist. Perhaps the classifier is training on some inconsequential artifact of the dataset. At the other extreme $g = 1.0$ suggests that the two confounding sets of six categories were completely indistinguishable. Such a classifier is not discriminating enough to differentiate between *airplanes* and the more specific category *fighter-jet*, or between *people* and their *faces*. In other words, the classifier generalizes so well about similar object classes that it may be considered too sloppy for some applications.

In practice the desired value of g depends on the needs of the customer. Lower values of g denote fine discrimination between similar categories or sub-categories. This would be particularly desirable in situations that require the exact identification of a particular species of mammal. A more inclusive classifier tends toward higher value of g. Such a classifier would presumably be better at identifying a mammal it has never seen before, based on general features shared

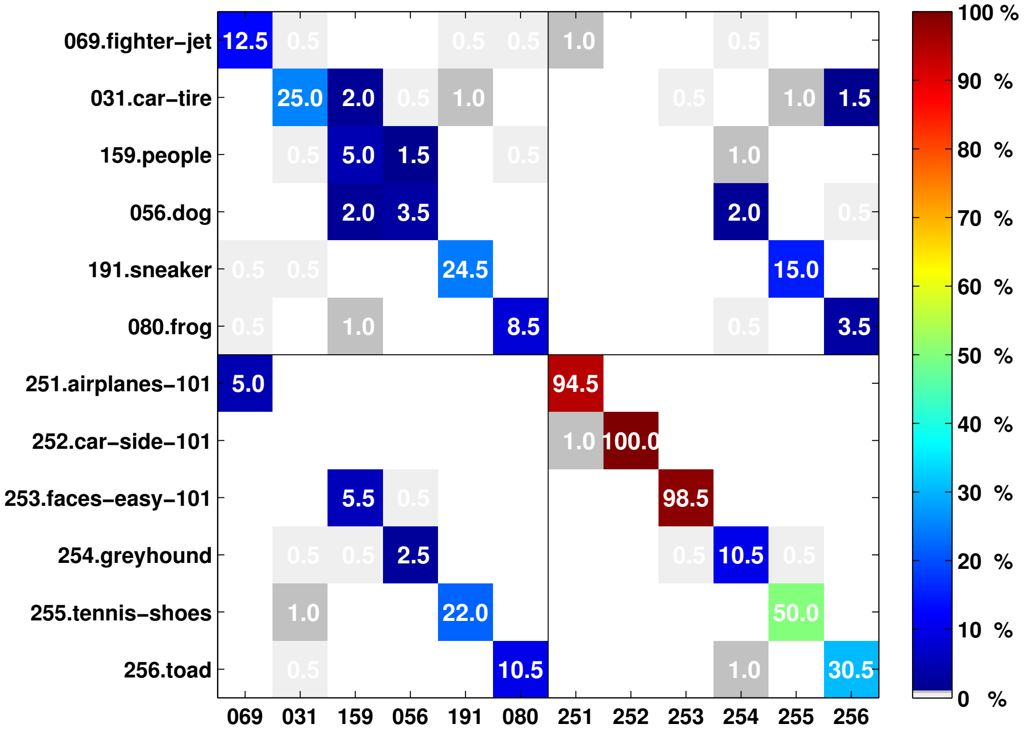

Figure 13: Selected rows and columns of the 256 × 256 confusion matrix $\mathcal{M}$ for spatial pyramid matching [2] and $N_{\text{train}} = 30$. Matrix elements containing 0.0 have been left blank. The first 6 categories are chosen because they are likely to be confounded with the last 6 categories. The main diagonal shows the performance for just these 12 categories. The diagonals of the other 2 quadrants show whether the algorithm can detect categories which are similar but not exact.

by a large class of mammals.

As shown in Figure 13, a spatial pyramid matching classifier does indeed confuse *tennis-shoes* and *sneakers* the most. This is a reassuring sanity check. To a lesser extent the object categories *frog*/*toad*, *dog*/*greyhound*, *fighter-jet*/*airplanes* and *people*/*faces-easy* are also confused.

Confusion between *car-tire* and *car-side* is entirely absent. This seems surprising since tires are such a conspicuous feature of cars when viewed from the side. However the tires pictured in *car-tire* tend to be much larger in scale than those found in *car-side*. One reasonable hypothesis is that the classifier has limited scale-invariance: objects or pieces of objects are no longer recognized if their size changes by an order of magnitude. This characteristic of the classifier may or may not be important, depending on the application. Another hypothesis is that the classifier relies not just on the presence of individual parts, but on their relationship to one another.

In short, generality defines a trade-off between classifier precision and robustness. Our metric for generating g is admittedly crude because it uses only six pairs of similar categories. Nonetheless generating a confusion matrix like the one shown in Figure 13 can provide a useful sanity check, while exposing features of a particular classifier that are not apparent from the raw performance benchmark.

4.5 Background

Returning to the example of a Mars rover, suppose that a small camera window is used to scan across the surface of the planet. Because there may be only one interesting object in 100 or 1000 images, the interest detector must have a low rate of false detections in order to be effective. As illustrated in figure 14 this is a challenging problem, particularly when the detector must accommodate hundreds of different object categories that are all considered *interesting*.

In the spirit of the attentional cascade [18] we train interest classifiers to discover which region are worthy of detailed classification and which are not. These detectors are summarized below. As before the classifier is an SVM with a spatial pyramid matching kernel [2]. The margin threshold is adjusted in order to trace out a full ROC curve[12].

Interest	N_{train}		*Speed*	*Description*
Detector	$\mathcal{C}_1 ... \mathcal{C}_{256}$	$\mathcal{C}_{257}$	*(images/sec)*	
A	30	512	24	Modified 257-category classifier
B	2	512	4600	Fast two-category classifier
C	30	30	25	Ordinary 257-category classifier

First let us consider *Interest Detector C*. This is the same detector that was employed for recognizing object categories in section 4.3. The only differences is that 257 categories are used instead of 256. Performance is poor because only 30 *clutter* images are used during training. In other words, *clutter* is treated exactly like any other category.

Interest Detector A corrects the above problem by using 512 training images from the *clutter* category. Performance improves because their is now a balance between the number of positive and negative examples. However the detector is still slow because it is a attempts to recognize 257 different object categories in every single image or camera region. This is wasteful if we expect the vast majority of regions to contain irrelevant clutter which is not worth classifying. In fact this detector only classifies about 25 images per second on a 3 GHz Pentium-based PC.

Interest Detector B trains on 512 *clutter* images and 512 images taken from the other 256 object categories. These two groups of images are assigned to the categories *uninteresting* and *interesting*, respectively. This *B* classifier is extremely fast because it combines all the *interesting* images into a single category instead of treating them as 256 separate categories. On a typical 3GHz Pentium processor this classifier can evaluate 4600 images (or scan regions) per second.

It may seem counter-intuitive to group two images from each category $\mathcal{C}_1 ... \mathcal{C}_{256}$ into a huge meta-category, as is done with Interest Detector B. What exactly is the classifier training on? What makes an image *interesting*? What if we have merely created a classifier that detects the photographic style of Stephen Shore? For these reasons any classifier which implements attention should be verified on a variety of background images, not just those in $\mathcal{C}_{257}$. For example the Caltech-6 provides 550 background images with very different statistics.

[12] When measuring speed, training time is ignored because it is a one-time expense

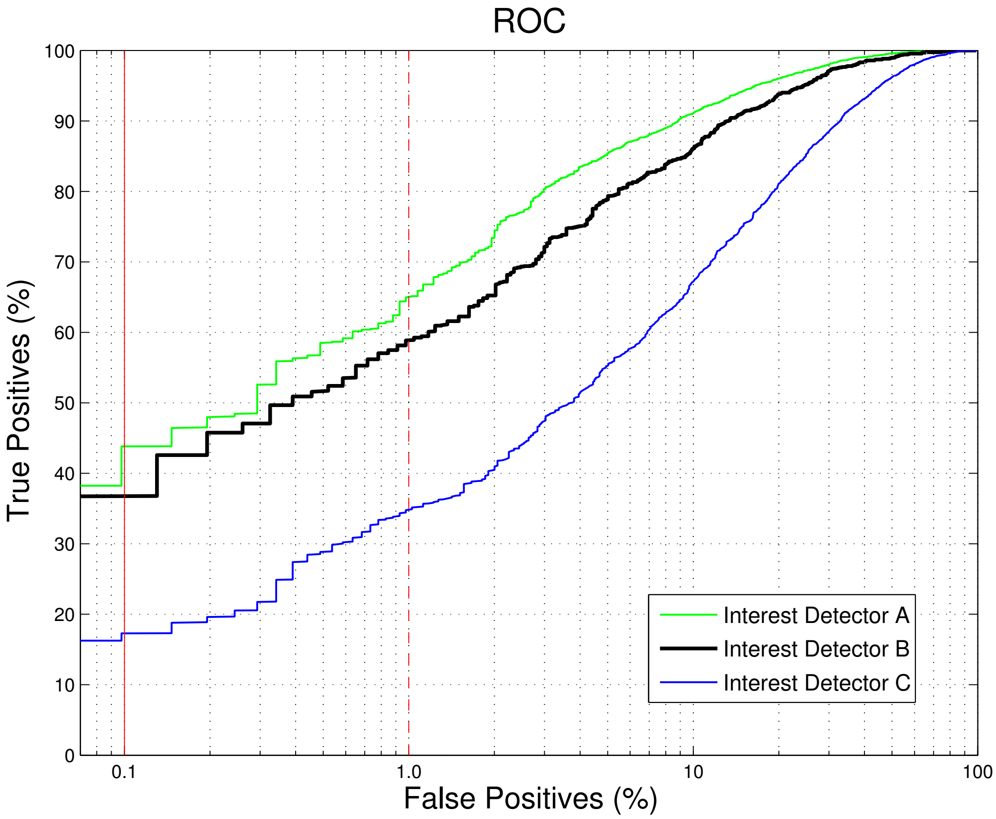

Figure 14: ROC curve for three different interest classifiers described in section 4.5. These classifiers are designed to focus the attention of the multi-category detectors benchmarked in Figure 12. Because *Detector B* is roughly 200 times faster than *A* or *C*, it represents the best tradeoff between performance and speed. This detector can accurately detect 38.2% of the interesting (non-clutter) images with a 0.1% rate of false detections. In other words, 1 in 1000 of the images classified as *interesting* will instead contain clutter (solid red line). If a 1 in 100 rate of false detections is acceptable, the accuracy increases to 58.6% (dashed red line).

5 Conclusion

Thanks to rapid advances in the vision community over the last few years, performance over 60% on the Caltech-101 has become commonplace. Here we present a new Caltech-256 image dataset, the largest set of object categories available to our knowledge. Our intent is to provide a freely available set of visual categories that does a better job of challenging today's state-of-the-art classification algorithms.

For example, spatial pyramid matching [2] with $N_{\text{train}} = 30$ achieves performance of 67.6% on the Caltech-101 as compared to 34.1% on Caltech-256. The standard practice among authors in the vision community is to benchmark raw classification performance as a function of training examples. As classification performance continues to improve, however, new benchmarks will be needed to reflect the performance of algorithms under realistic conditions. Beyond raw performance, we argue that a successful algorithm should also be able to

- Generalize beyond a specific set of images or categories
- Identify which images or image regions are worth classifying

In order to evaluate these characteristics we test two new benchmarks in the context of Caltech-256. No doubt there are other equally relevant bench-

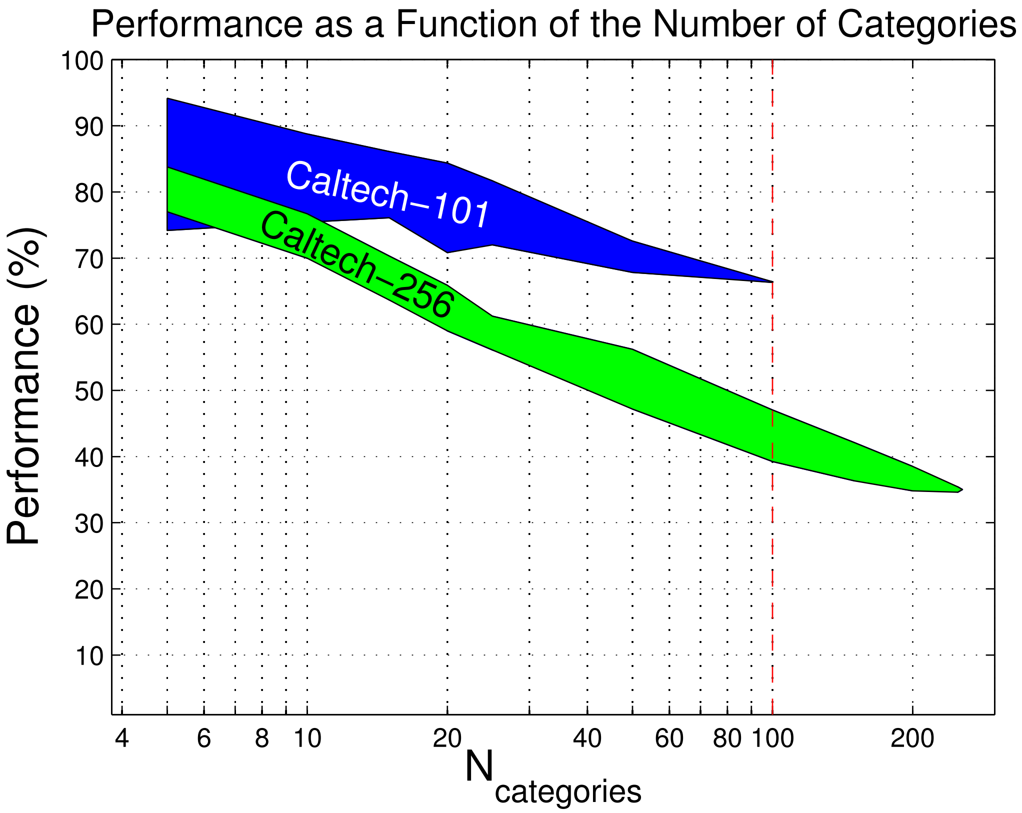

Figure 15: In general the Caltech-256 images are more difficult to classify than the Caltech-101 images. Here we plot performance of the two datasets over a random mix of $N_{\text{categories}}$ from each dataset. Even when the number of categories remains the same, the Caltech-256 performance is lower. For example at $N_{\text{categories}} = 100$ the performance is $\sim 60\%$ lower.

marks that we have not considered. We invite researchers to devise suitable benchmarks and share them with the community at large.

If you would like to share performance results as well as your confusion matrix, please send them to `caltech256@vision.caltech.edu`. We will try to keep our comparison of performance as up-to-date as possible. For more details see

`http://www.vision.caltech.edu/Image_Datasets/Caltech256`

6 Acknowledgments

We are indebted to Marco Andreetto, Fei Fei Li and Marc'Aurelio Ranzato who collected Caltech-101. We also made use of some parts of the Caltech-101 code written by Rob Fergus. Pierre Moreels provided code and guidance. Finally thanks to our sorters Elisabeth Fano, Nick Lo, Julie May and Weiyu Xu for their diligent work. Marco Andreetto and Claudio Fanti also sorted images.

References

[1] F.F. Li, R. Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004.

[2] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2006.

[3] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2006.

[4] Jim Mutch and David G. Lowe. Multiclass object recognition with sparse, localized features. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2006.

[5] R. Fergus. *Visual Object Category Recognition.* PhD thesis, University of Oxford, 2005.

[6] Tamara L. Berg Alexander Berg and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. Technical Report UCB/CSD-04-1366, EECS Department, University of California, Berkeley, 2004.

[7] S. Nene, S. Nayar, and H. Murase. Columbia object image library: Coil, 1996.

[8] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 762–769, Washington, DC, June 2004.

[9] Everingham and Mark et al. The 2005 pascal visual object classes challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment (PASCAL Workshop 05)*, number 3944 in Lecture Notes in Artificial Intelligence, pages 117–176, Southampton, UK, 2006.

[10] A. Opelt and A. Pinz. Object localization with boosting and weak supervision for generic object recognition. In *Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA)*, 2005.

[11] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *MIT AI Lab Memo AIM-2005-025*, 2005.

[12] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[13] Stephen Shore. *Uncommon Places: The Complete Works.* Aperture, 2004.

[14] Stephen Shore. *American Surfaces.* Phaidon Press, 2005.

[15] Alex Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *ICCV*, pages 136–143, 2005.

[16] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[17] Alexander C. Berg, Tamara L. Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. *cvpr*, 1:26–33, 2005.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001.