80 million tiny images: a large dataset for non-parametric object and scene recognition

Antonio Torralba, Rob Fergus and William T. Freeman

***Abstract*— With the advent of the Internet, billions of images are now freely available online and constitute a dense sampling of the visual world. Using a variety of non-parametric methods, we explore this world with the aid of a large dataset of 79,302,017 images collected from the Internet.**

Motivated by psychophysical results showing the remarkable tolerance of the human visual system to degradations in image resolution, the images in the dataset are stored as 32 × 32 color images. Each image is loosely labeled with one of the 75,062 non-abstract nouns in English, as listed in the Wordnet lexical database. Hence the image database gives a comprehensive coverage of all object categories and scenes. The semantic information from Wordnet can be used in conjunction with nearest-neighbor methods to perform object classification over a range of semantic levels minimizing the effects of labeling noise. For certain classes that are particularly prevalent in the dataset, such as people, we are able to demonstrate a recognition performance comparable to class-specific Viola-Jones style detectors. We also demonstrate a range of other applications of this very large dataset including automatic image colorization and picture orientation determination.

***Index Terms*— Object recognition, tiny images, large datasets, Internet images, nearest-neighbor methods.**

I. Introduction

With overwhelming amounts of data, many problems can be solved without the need for sophisticated algorithms. One example in the textual domain is Google's "Did you mean?" tool which corrects errors in search queries, not through a complex parsing of the query but by memorizing billions of query-answer pairs and suggesting the one closest to the users query. In this paper, we explore a visual analog to this tool by using a large dataset of 79 million images and nearest-neighbor matching schemes.

When very many images are available, simple image indexing techniques can be used to retrieve images with object arrangements to the query image. If we have a big enough database then we can find, with high probability, images visually close similar to a query image, containing similar scenes with similar objects arranged in similar spatial configurations. If the images in the retrieval set are partially labeled, then we can propagate the labels to the query image, so performing classification.

Nearest-neighbor methods have been used in a variety of computer vision problems, primarily for interest point matching [5], [17], [27]. They have also been used for global image matching (e.g. estimation of human pose [36]), character recognition [3], and object recognition [5], [35]. A number of recent papers have used large datasets of images in conjunction with purely non-parametric methods for computer vision and graphics applications [20], [39].

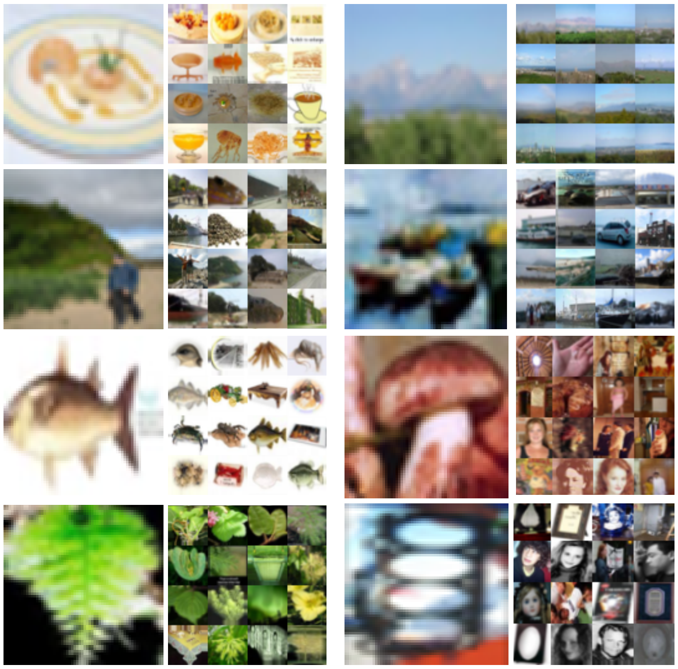


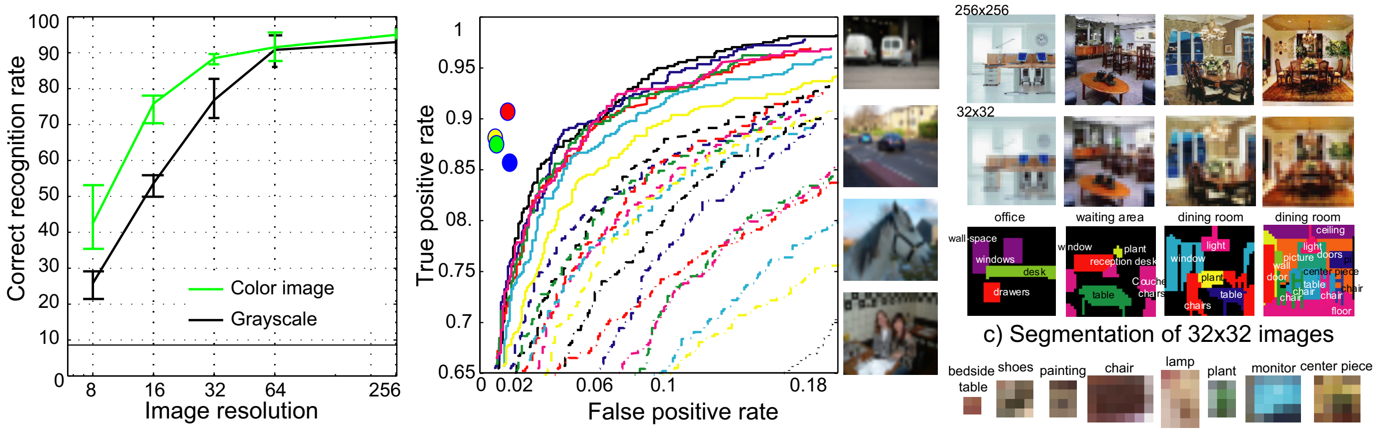
Fig. 1. 1st & 3rd columns: Eight 32 × 32 resolution color images. Despite their low resolution, it is still possible to recognize most of the objects and scenes. These are samples from a large dataset of 10^8 32 × 32 images we collected from the web which spans all visual object classes. 2nd & 4th columns: Collages showing the 16 nearest neighbors within the dataset to each image in the adjacent column. Note the consistency between the neighbors and the query image, having related objects in similar spatial arrangements. The power of the approach comes from the copious amount of data, rather than sophisticated matching methods.

The key question that we address in this paper is: How big does the image dataset need to be to robustly perform recognition using simple nearest-neighbor schemes? In fact, it is unclear that the size of the dataset required is at all practical since there are an effectively infinite number of possible images the visual system can be confronted with. What gives us hope is that the visual world is very regular in that real world pictures occupy only a relatively small portion of the space of possible images.

Studying the space occupied by natural images is hard due to the high dimensionality of the images. One way of simplifying this task is by lowering the resolution of the images. When we look at the images in Fig. 1, we can recognize the scene and its constituent objects. Interestingly though, these pictures have only 32×32 color pixels (the entire image is just a vector of 3072 dimensions with 8 bits per dimension), yet at this resolution, the images already seem to contain most of the relevant information needed to support reliable recognition.

An important benefit of working with tiny images is that it becomes practical to store and manipulate datasets orders of

The authors are with the Computer Science and Artificial Intelligence Lab (CSAIL) at the Massachusetts Institute of Technology.
Email: {torralba,fergus,billf}@csail.mit.edu



a) Scene recognition

b) Car detection

c) Segmentation of 32x32 images

d) Cropped objects

Fig. 2. a) Human performance on scene recognition as a function of resolution. The green and black curves show the performance on color and grayscale images respectively. For color 32 × 32 images the performance only drops by 7% relative to full resolution, despite having 1/64th of the pixels. b) Car detection task on the PASCAL 2006 test dataset. The colored dots show the performance of four human subjects classifying tiny versions of the test data. The ROC curves of the best vision algorithms (running on full resolution images) are shown for comparison. All lie below the performance of humans on the tiny images, which rely on none of the high-resolution cues exploited by the computer vision algorithms. c) Humans can correctly recognize and segment objects at very low resolutions, even when the objects in isolation can not be recognized (d).

magnitude bigger than those typically used in computer vision. Correspondingly, we introduce, and make available to researchers, a dataset of 79 million unique 32×32 color images gathered from the Internet. Each image is loosely labeled with one of 75,062 English nouns, so the dataset covers all visual object classes. This is in contrast to existing datasets which provide a sparse selection of object classes.

The paper is divided in three parts. In Section 2 we investigate the limits of human recognition, establishing the minimal resolution required for scene and object recognition. In Sections 3 and 4 we introduce our dataset of 79 million images and explore some of its properties. In Section 5 we attempt scene and object recognition using a variety of nearest-neighbor methods. We measure performance at a number of semantic levels, obtaining impressive results for certain object classes.

II. Low dimensional image representations

Non-parametric approaches must cover the input space, and our scheme relies on the dataset of 79 million images densely populating the manifold of natural images. We seek a compact image representation in which the intrinsic dimensionality of the manifold is a low as possible, since that makes the manifold easy to cover, while preserving the semantic content. One of the simplest mechanisms to reduce the dimensionality of an image is by lowering its resolution. A second benefit of a low resolution representation is that the images can be indexed efficiently and provide the storage savings essential for dealing with very large datasets. However, it is important that the low dimensional representation not loses important image information. In this section we study the minimal image resolution which still retains useful information about the visual world. In order to do this, we perform a series of human experiments on (i) scene recognition and (ii) object recognition. Studies on face perception [1], [19] have shown that only 16×16 pixels are needed for robust face recognition. This remarkable performance is also found in a scene recognition task [31].

In this section we provide experimental evidence showing that 32×32 color images[1] contain enough information for scene recognition, object detection and segmentation (even when the objects occupy just a few pixels in the image). As we will see in Fig. 2, a significant drop in performance is observed when the resolution drops below 32^2 pixels. Note that this problem is distinct from studies investigating scene recognition using very short presentation times [11], [30], [33], [34]. Here, we are interested in characterizing the amount of information available in the image as a function of the image resolution (with no constraint on presentation time).

In cognitive psychology, the *gist* of the scene [30], [44] refers to a short summary of the scene (the scene category, and a description of a few objects that compose the scene). In computer vision, the term *gist* is used to refer to a low dimensional representation of the entire image. Low dimensional global image representation have been used to for scene recognition [16], [32], [22], for providing context for object detection [38], [40], depth estimation [41] and image retrieval for computer graphics [20]. In this section, we show that this low dimensional representation can rely on very low-resolution information and, therefore, can be computed very efficiently.

A. Scene recognition

We evaluate the scene recognition performance of humans as the image resolution is decreased. We used a dataset of 15 scenes was taken from [12], [22], [32]. Each image was shown at one of 5 possible resolutions (8^2, 16^2, 32^2, 64^2 and 256^2 pixels) and the participant task was to assign the low-resolution picture to one of the 15 different scene categories (bedroom, suburban, industrial, kitchen, living room, coast, forest, highway, inside city,

[1] 32×32 is very very small. For reference, typical thumbnail sizes are: Google images (130×100), Flikr (180×150), default Windows thumbnails (90×90).

mountain, open country, street, tall buildings, office, and store)[2]. Fig. 2(a) shows human performance on this task when presented with grayscale and color images[3] of varying resolution. For grayscale images, humans need around 64×64 pixels. When the images are in color, humans need only 32×32 pixels. Below this resolution the performance rapidly decreases. Therefore, humans need around 3000 dimensions of either color or grayscale data to perform this task. In the next section we show that 32×32 color images also preserve a great amount of local information and that many objects can still be recognized even when they occupy just a few pixels.

B. Object recognition

Recently, the PASCAL object recognition challenge evaluated a large number of algorithms in a detection task for several object categories [10]. Fig. 2(b) shows the performances (ROC curves) of the best performing algorithms in the car classification task (i.e. is there a car present in the image?). These algorithms require access to relatively high resolution images. We studied the ability of human participants to perform the same detection task but using very low-resolution images. Human participants were shown color images from the test set scaled to have 32 pixels on the smallest axis, preserving their aspect ratio. Fig. 2(b) shows some examples of tiny PASCAL images. Each participant classified between 200 and 400 images selected randomly. Fig. 2(b) shows the performances of four human observers that participated in the experiment. Although around 10% of cars are missed, the performance is still very good, significantly outperforming the computer vision algorithms using full resolution images. This shows that even though the images are very small, they contain sufficient information for accurate recognition.

Fig. 2(c) shows some representative 32^2 images segmented by human subjects. It is important to note that taking objects out of their context drastically reduces recognition rate. Fig. 2(d) shows crops of some of the smallest objects correctly recognized when shown within the scene. Note that in isolation, the objects cannot be identified since the resolution is so low. Hence the recognition of these objects within the scene is almost entirely based on context. Clearly, sufficient information remains for reliable segmentation. However, not all visual tasks can be solved using such low resolution images. The experiments in this section have studied only recognition tasks — the focus of this paper. The results in this section suggest that 32×32 color images are the minimum viable size at which to study the manifold of natural images. Any further lowering in resolution results in a rapid drop in recognition performance.

III. A LARGE DATASET OF 32×32 IMAGES

As discussed in the previous sections, 32×32 color images contain the information needed to perform a number of challenging recognition tasks. One important advantage of very low resolution images is that it becomes practical to work with millions of images. In this section we will describe a dataset of 10^8 tiny images.

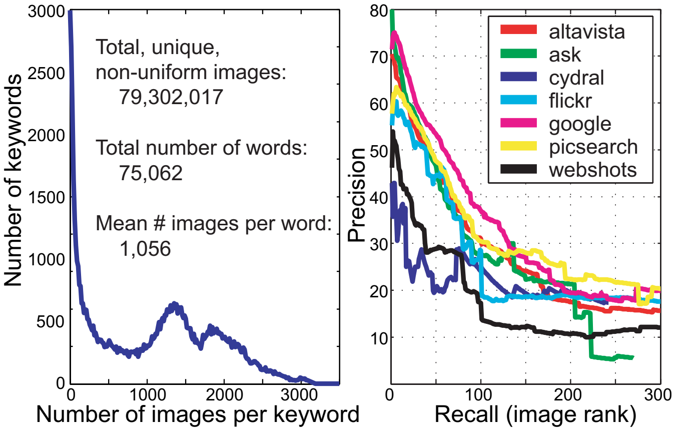


Fig. 3. Statistics of the tiny images database. a) A histogram of images per keyword collected. Around 10% of keywords have very few images. b) Performance of the various engines (evaluated on hand-labeled ground truth). Google and Altavista are the best performing while Cydral and Flickr the worst.

Current experiments in object recognition typically use 10^2-10^4 images spread over a few different classes; the largest available dataset being one with 256 classes from the Caltech vision group [18]. Other fields, such as speech, routinely use 10^6 data points for training, since they have found that large training sets are vital for achieving low errors rates in testing. As the visual world is far more complex than the aural one, it would seem natural to use very large set of training images. Motivated by this, and the ability of humans to recognize objects and scenes in 32×32 images, we have collected a database of nearly 10^8 such images, made possible by the minimal storage requirements for each image.

A. Collection procedure

We use Wordnet[4] to provide a comprehensive list of all classes[5] likely to have any kind of visual consistency. We do this by extracting all non-abstract nouns from the database, 75,062 of them in total. In contrast to existing object recognition datasets which use a sparse selection of classes, by collecting images for all nouns, we have a dense coverage of all visual forms.

We selected 7 independent image search engines: Altavista, Ask, Flickr, Cydral, Google, Picsearch and Webshots (others have outputs correlated with these). We automatically download all the images provided by each engine for all 75,846 non-abstract nouns. Running over 8 months, this method gathered 97,245,098 images in total. Once intra-word duplicates[6] and uniform images (images with zero variance) are removed, this number is reduced to 79,302,017 images from 75,062 words (around 1% of the

[2] Experimental details: 6 participants classified 585 color images as belonging to one of the 15 scene categories from [12], [22], [32]. Images were presented at 5 possible resolutions (8^2, 16^2, 32^2, 64^2 and 256^2). Each image was shown at 5 possible sizes using bicubic interpolation to reduce pixelation effects which impair recognition. Interpolation was applied to the low-resolution image with 8 bits per pixel and color channel. Images were not repeated across conditions. 6 additional participants performed the same experiment but with gray scale images.

[3] 100% recognition rate can not be achieved in this dataset as there is no perfect separation between the 15 categories.

[4] Wordnet [13] is a lexical dictionary, meaning that it gives the semantic relations between words in addition to the information usually given in a dictionary.

[5] The tiny database is not just about objects. It is about everything that can be indexed with Wordnet and this includes scene-level classes such as streets, beaches, mountains, as well as category-level classes and more specific objects such as US Presidents, astronomical objects and Abyssinian cats.

[6] At present we do not remove inter-word duplicates since identifying them in our dataset is non-trivial.

keywords had no images). Storing this number of images at full resolution is impractical on the standard hardware used in our experiments so we down-sampled the images to 32×32 as they were gathered[7]. The dataset fits onto a single hard disk, occupying 760Gb in total. The dataset may be downloaded from `http:\\people.csail.mit.edu\torralba\tinyimages`.

Fig. 3(a) shows a histogram of the number of images per class. Around 10% of the query words are obscure so no images can be found on the Internet, but for the majority of words a reasonable number of images are found. We place an upper limit of 3000 images/word to keep the total collection time to a reasonable level. Although the gathered dataset is very large, it is not necessarily representative of all natural images. Images on the Internet have their own biases (e.g. objects tend to be centered and fairly large in the image). However, web images define an interesting visual world for developing computer vision applications [14], [15], [37].

B. *Characterization of labeling noise*

The images gathered by the engines are loosely labeled in that the visual content is often unrelated to the query word (for example, see Fig. 13). In Fig. 3(b) we quantify this using a hand-labeled portion of the dataset. 78 animal classes were labeled in a binary fashion (belongs to class or not) and a recall-precision curve was plotted for each search engine. The differing performance of the various engines is visible, with Google and Altavista performing the best and Cydral and Flickr the worst. Various methods exist for cleaning up the data by removing images visually unrelated to the query word. Berg and Forsyth [6] have shown a variety of effective methods for doing this with images of animals gathered from the web. Berg et al. [4] showed how text and visual cues could be used to cluster faces of people from cluttered news feeds. Fergus et al. [14], [15] have shown the use of a variety of approaches for improving Internet image search engines. Li et al. [25] show further approaches to decreasing label noise. However, due to the extreme size of our dataset, it is not practical to employ these methods. In Section 5, we show that reasonable recognition performances can be achieved despite the high labeling noise.

IV. Statistics of very low resolution images

Despite 32×32 being very low resolution, each image lives in a space of 3072 dimensions. This is a very large space — if each dimension has 8 bits, there are a total of 10^{7400} possible images. This is a huge number, especially if we consider that a human in a 100 years only gets to see 10^{11} frames (at 30 frames/second).

However, natural images only correspond to a tiny fraction of this space (most of the images correspond to white noise), and it is natural to investigate the size of that fraction. A number of studies [8], [23] have been devoted to characterize the space of natural images by studying the statistics of small image patches. However, low-resolution scenes are quite different to patches extracted by randomly cropping small patches from images.

Given a similarity measure, the question that we want to answer is: *how many images are needed to be able to find a similar image to match any imput image?* As we increase the size of the dataset, the probability of finding similar images will increase. The goal of this section is to evaluate experimentally how fast this probability increases as a function of dataset size. In turn, this tells us how big the dataset needs to be to give a robust recognition performance.

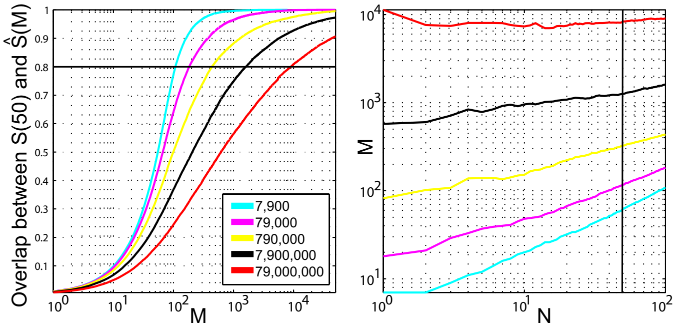


Fig. 4. Evaluation of the method for computing approximate nearest neighbors. (a) Probability that an image from the set of exact nearest neighbors S_N , with $N = 50$, is inside the approximate set of nearest neighbors $\hat{S}_M$ as a function of M. b) Number of approximate neighbors (M) that need to be considered as a function of the desired number of exact neighbors (N) in order to have a probability of 0.8 of finding N exact neighbors. Each graph corresponds to a different dataset size, indicated by the color code.

A. *Distribution of neighbors as a function of dataset size*

As a first step, we use the sum of squared differences (SSD) to compare two images. We will define later other similarity measures that incorporate invariances to translations and scaling. The SSD between two images I_1 and I_2 is:

$$D^2_{\text{ssd}} = \sum_{x,y,c} (I_1(x,y,c) - I_2(x,y,c))^2 \qquad (1)$$

Each image is normalized to have zero mean and unit norm[8]. Computing similarities among 7.9×10^7 images is computationally expensive. To improve speed, we index the images using the first 19 principal components of the 7.9×10^7 images (19 is the maximum number of components per image such that the entire index structure can be held in memory). The $1/f^2$ property of the power spectrum of natural images means that the distance between two images can be approximated using few principal components. We compute the approximate distance $\hat{D}^2_{\text{ssd}} = \sum_{n=1}^{C} (v_1(n) - v_2(n))^2$, where $v_i(n)$ is the n^{th} principal component coefficient for the i^{th} image, and C is the number of components used to approximate the distance. We define S_N as the set of N exact nearest neighbors and $\hat{S}_M$ as the set of M approximate nearest neighbors. Fig. 4(a) shows the probability that an image, of index i, from the set S_N is also inside $\hat{S}_M$: $P(i \in \hat{S}_M | i \in S_N)$. The plot corresponds to $N = 50$. Fig. 4(b) shows the number of approximate neighbors (M) that need to be considered as a function of the desired number of exact neighbors (N) in order to have a probability of 0.8 of finding N exact neighbors. As the dataset becomes larger, we need to collect more approximate nearest neighbors in order to have the same probability of including the first N exact nearest neighbors. These plots were obtained from 200 images for which we computed the exact distances to all the 7.9×10^7 images.

For the experiments in this paper, we use the following procedure. First, using exhaustive search we find the closest 16,000

[7] We store a version of the images that maintained the original aspect ratio (the minimum dimension was set at 32 pixels) and a link to the original thumbnail and high resolution URL.

[8] Normalization of each image is performed by transforming the image into a vector concatenating the three color channels. The normalization does not change image color, only the overall luminance.

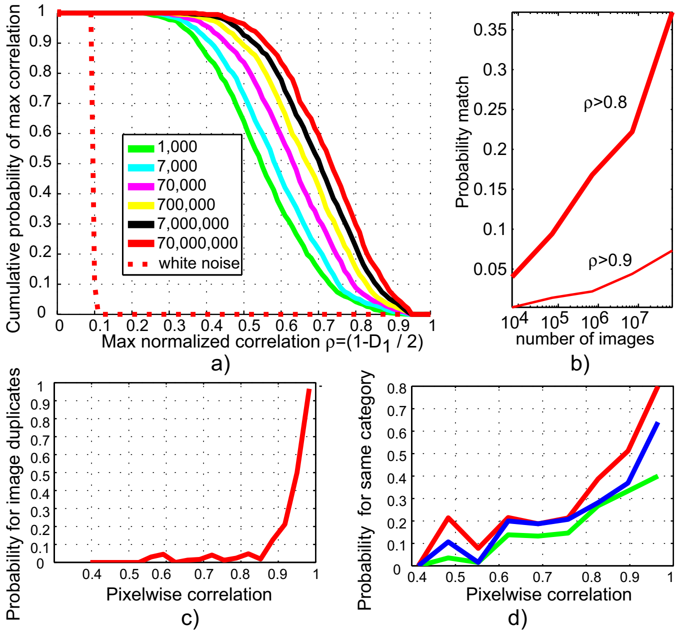


Fig. 5. Exploring the dataset using D_{ssd}. (a) Cumulative probability that the nearest neighbor has a correlation greater than ρ. Each of the colored curves shows the behavior for a different size of dataset. (b) Cross-section of figure (a) plots the probability of finding a neighbor with correlation > 0.9 as a function of dataset size. (c) Probability that two images are duplicates as a function of pixel-wise correlation. (d) Probability that two images belong to the same category as a function of pixel-wise correlation (duplicate images are removed). Each curve represents a different human labeler.

images[9] per image. From Fig. 4(a) we know that more than 80% of the exact neighbors will be part of this approximate neighbor set. Then, within the set of 16,000 images, we compute the exact distances to provide the final rankings of neighbors.

Fig. 5 shows several plots measuring various properties as the size of the dataset is increased. The plots use the normalized correlation ρ between images (note that $D^2_{\text{ssd}} = 2(1-\rho)$). In Fig. 5(a), we show the probability that the nearest neighbor has a normalized correlation exceeding a certain value. Each curve corresponds to a different dataset size. Fig. 5(b) shows a vertical section through Fig. 5(a) at the correlations 0.8 and 0.9, plotting the probability of finding a neighbor as the number of images in the dataset grows. ¿From Fig. 5(b) we see that a third of the images in the dataset are expected to have a neighbor with correlation > 0.8.

Many images on the web appear multiple times. Fig. 5(c) shows the probability of the matched image being a duplicate as a function of D_{ssd}. For the other plots in this figure, we have removed manually all the image pairs that were duplicates.

In Fig. 5(d) we explore how the plots shown in Fig. 5(a) & (b) relate to recognition performance. Three human subjects labeled pairs of images as belonging to the same visual class or not (pairs of images that correspond to duplicate images are removed). The plot shows the probability that two images are labeled as belonging to the same class as a function of image similarity.

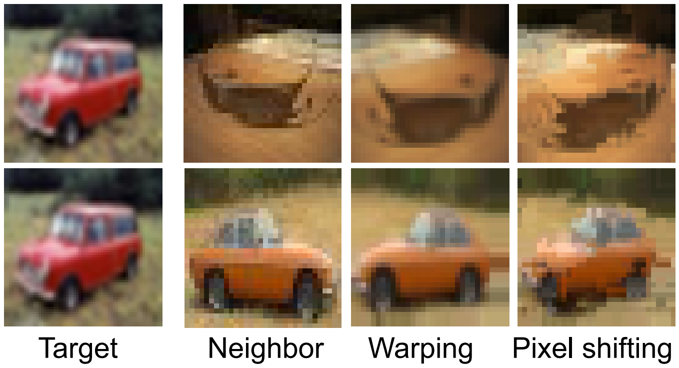


Fig. 6. Image matching using distance metrics D_{ssd}, D_{warp} and D_{shift}. After transforming each neighbor by the optimal transformation; the sunglasses always results in a poor match. However, for the car example, the matched image approximates the pose of the target car.

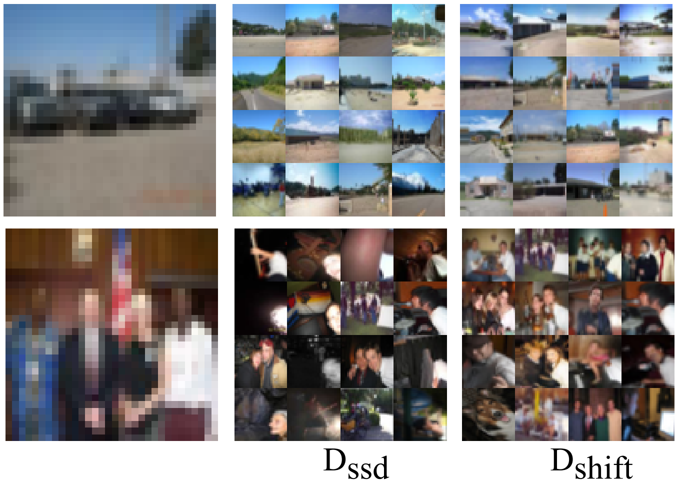


Fig. 7. Sibling sets from 79,302,017 images, found with distance metrics D_{ssd}, and D_{shift}. D_{shift} provides better matches than D_{ssd}.

As the normalized correlation exceeds 0.8, the probability of belonging to the same class grows rapidly. Hence a simple K-nearest-neighbor approach might be effective with our size of dataset. We will explore this further in Section V.

B. *Image similarity metrics*

We can improve recognition performance using better measures of image similarity. We now introduce two additional similarity measures between a pair of images I_1 and I_2, that incorporate invariances to simple spatial transformations.

- In order to incorporate invariance to small translations, scaling and image mirror, we define the similarity measure:

  $$D^2_{\text{warp}} = \min_{\theta} \sum_{x,y,c} (I_1(x,y,c) - T_\theta[I_2(x,y,c)])^2$$

  In this expression, we minimize the similarity by transforming I_2 (horizontal mirror; translations and scaling up to 10 pixel shifts) to give the minimum SSD. The transformation parameters θ are optimized by gradient descent [28].
- We allow for additional distortion in the images by shifting every pixel individually within a 5 by 5 window to give minimum SSD. This registration can be performed with

[9] The exhaustive search currently takes 30 seconds per image using the principle components method. Undoubtedly, if efficient data structures such as a kd-tree were used, the matching would be significantly faster. Nister and Stewenius [29] used related methods to index over 1 million images in $\sim$ 1sec.

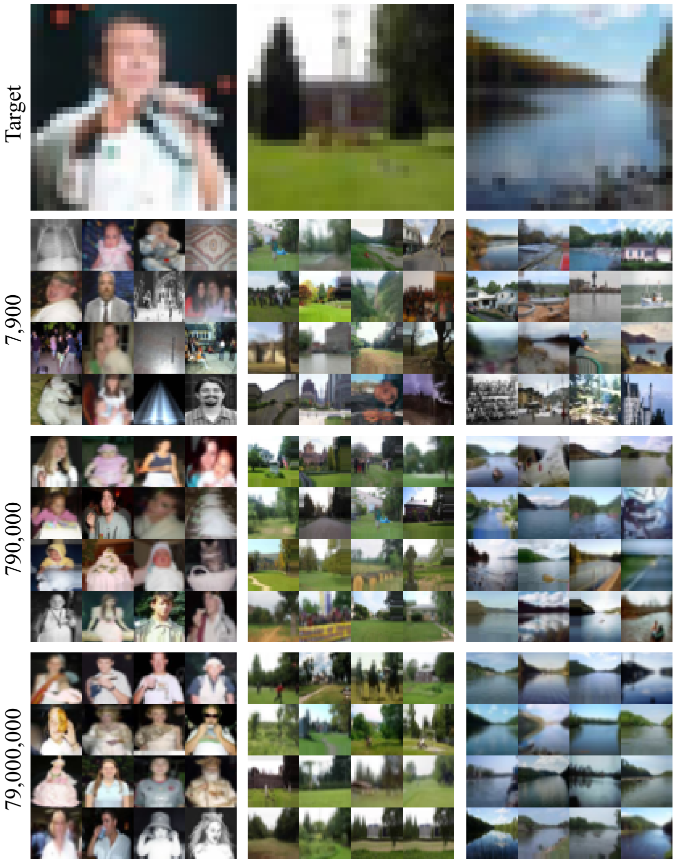


Fig. 8. As we increase the size of the dataset, the quality of the retrieved set increases dramatically. However, note that we need to increase the size of the dataset logarithmically in order to have an effect. These results are obtained using D_{shift} as a similarity measure between images.

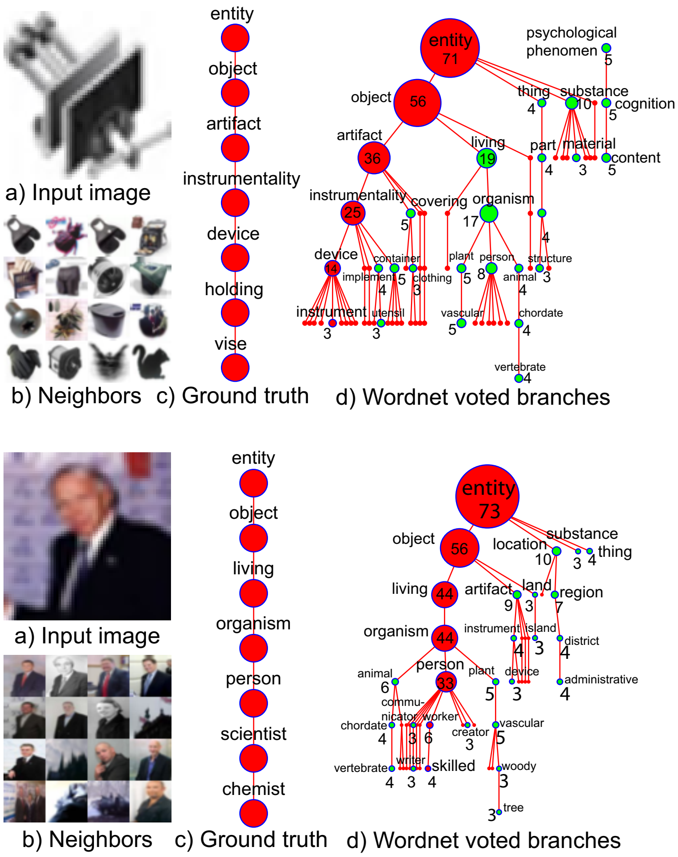


Fig. 9. This figure shows two examples. (a) Query image. (b) First 16 of 80 neighbors found using D_{shift}. (c) Ground truth Wordnet branch describing the content of the query image at multiple semantic levels. (d) Sub-tree formed by accumulating branches from all 80 neighbors. The number in each node denotes the accumulated votes. The red branch shows the nodes with the most votes. Note that this branch substantially agrees with the branch for vise and for person in the first and second examples respectively.

more complex representations than pixels (e.g., Berg and Malik [5]). In our case, the minimum can be found by exhaustive evaluation of all shifts, only possible due to the low resolution of the images.

$$D^2_{\text{shift}} = \min_{|D_{x,y}| \leq w} \sum_{x,y,c} (I_1(x,y,c) - \hat{I}_2(x+D_x, y+D_y, c))^2$$

In order to get better matches, we initialize I_2 with the warping parameters obtained after optimization of D_{warp}, $\hat{I}_2 = T_\theta[I_2]$.

Fig. 6 shows a pair of images being matched using the 3 metrics and shows the resulting neighbor images transformed by the optimal parameters that minimize each similarity measure. The figure shows two candidate neighbors: one matching the target semantic category and another one that corresponds to a wrong match. For D_{warp} and D_{shift} we show the closest manipulated image to the target. D_{warp} looks for the best translation, scaling and horizontal mirror of the candidate neighbor in order to match the target. D_{shift} further optimizes the warping provided by D_{warp} by allowing pixels to move independently in order to minimize the distance with the target. Fig. 7 shows two examples of query images and the retrieved sibling set, out of 79,302,017 images, using D_{ssd} and D_{shift}. Both measures provide very good matches, but D_{shift} returns closer images at the semantic level. This observation will be quantified in Section V.

Fig. 1 shows examples of query images and sets of neighboring images, from our dataset of 79,302,017 images, found using D_{shift}. In the rest of the paper we will call the set of neighboring images a *sibling set*. Fig. 8 shows the effects of increasing the dataset size on the quality of the sibling set. As we increase the size of the dataset, the quality of the retrieved set increases dramatically. Specifically, note the change in performance when using only around 10,000 images (a typical number used in image retrieval research) compared to 10^8. Despite the simplicity of the similarity measures used in these experiments, due to the large size of our dataset, the retrieved images are very similar (hence *siblings*) to the target image. We will now quantify this observation in the next section.

V. Recognition

A. *Wordnet voting scheme*

We now attempt to use our dataset for object and scene recognition. While an existing computer vision algorithm could be adapted to work on 32×32 images, we prefer to use a simple nearest-neighbor scheme based on one of the distance metrics D_{ssd}, D_{warp} or D_{shift}. Instead of relying on the complexity of the matching scheme, we let the data to do the work for us: the hope is that there will always be images close to a given

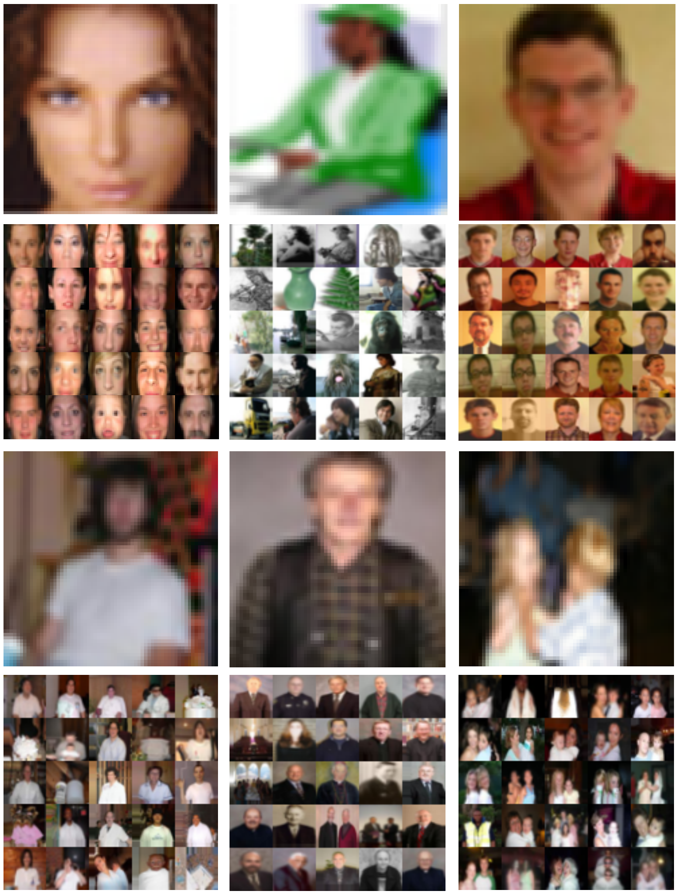


Fig. 10. Some examples of test images belonging to the “person” node of the Wordnet tree, organized according to body size. Each pair shows the query image and the 25 closest neighbors out of 79 million images using D_{shift} with 32×32 images. Note that the sibling sets contain people in similar poses, with similar clothing to the query images.

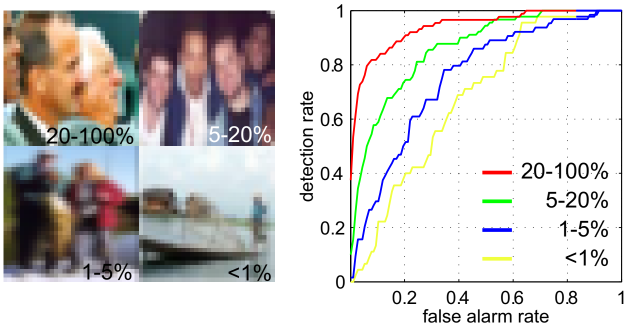


Fig. 11. ROC curves for people detection (not localization) in images drawn randomly from the dataset of 79 million. The performance is a function of the person’s size in an image, the numbers indicating the fraction of the image occupied by the head.

query image with some semantic connection to it. The goal of this section is to show that the performance achieved can match that of sophisticated algorithms which use much smaller training sets.

An additional factor in our dataset is the labeling noise. To cope with this we propose a voting scheme based around the Wordnet semantic hierarchy.

Wordnet [13] provides semantic relationships between the 75,062 nouns for which we have collected images. For simplicity, we reduce the initial graph-structured relationships between words to a tree-structured one by taking the most common meaning of each word. The result is a large semantic tree whose nodes consist of the 75,062 nouns and their hypernyms, with all the leaves being nouns[10]. Fig. 9(c) shows the unique branch of this tree belonging to the nouns “vise” and “chemist”. The Wordnet tree provides two benefits:

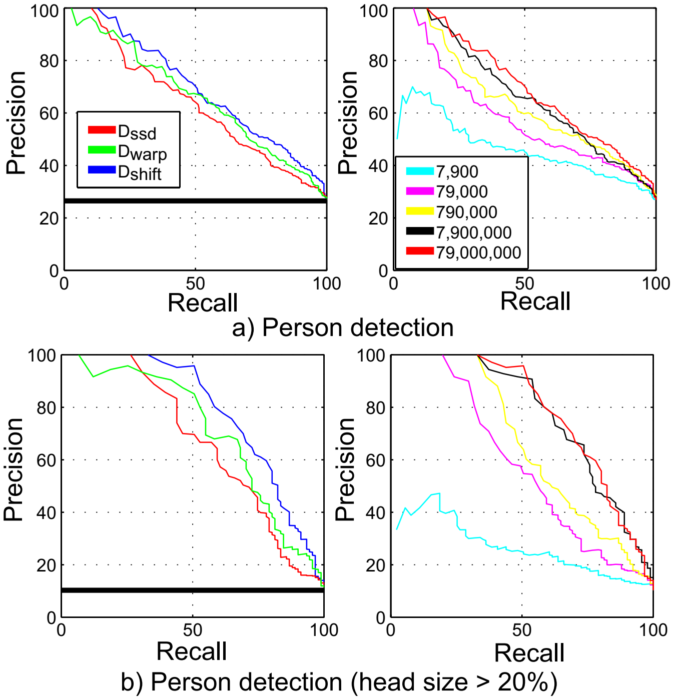
- Recognition of a test image can be performed at multiple semantic levels. Given the large number of classes in our dataset (75,062) and their highly specific nature, it is not practical or desirable to try and classify each of the classes separately. Instead, using the Wordnet hierarchy, we can perform classification at a variety of different semantic levels. So instead of just trying to recognize the noun “yellowfin tuna”, we may also perform recognition at the level of “tuna” or “fish” or “animal”. This is in contrast to current approaches to recognition that only consider a single, manually imposed, semantic meaning of an object or scene.
- If classification is performed at some intermediate semantic level, for example using the noun “person”, we need not only consider images gathered from the Internet using “person”. Using the Wordnet hierarchy tree, we can also draw on all images belonging to nouns whose hypernyms include “person” (for example, “arithmetician”). Hence, we can massively increase the number of images in our training set at higher semantic levels. Near the top of the tree, however, the nouns are so generic (e.g. “object”) that the child images recruited in this manner have little visual consistency, so despite their extra numbers may be of little use in classification[11].

Our classification scheme uses the Wordnet tree in the following way. Given a query image, the neighbors are found using some similarity measure. Each neighbor in turn votes for its branch within the Wordnet tree. In this manner votes are accumulated across a range of semantic levels and the effects of the labeling noise are averaged out over many neighbors. Classification may be performed by assigning the query image the label with the most votes at the desired height (i.e. semantic level) within the tree, the number of votes acting as a measure of confidence in the decision.

In Fig. 9(a) we show a query image of a vise from our test set. In Fig. 9(b) we show the first 16 images from the $K = 80$ nearest neighbors using D_{shift} over the 79 million images. Note that many of the neighbors, despite not being vices, are some kind of device or instrument. In Fig. 9(c) we show the Wordnet branch

[10] Note that not all nouns are leaf nodes since many of the 75,062 nouns are hypernyms of other nouns. E.g. “yellowfin tuna” and “fish” are two nouns. The former is a leaf node, while the latter is in intermediate node since “fish” is a hypernym of “yellowfin tuna”.

[11] The use of Wordnet tree in this manner implicitly assumes that semantic and visual consistency are tightly correlated. While this might be the case for certain nouns (for example, “poodle” and “dachshund”), it is not clear how true this is in general. To explore this issue, we constructed a poster consisting of 75,062 tiles. Each title is the arithmetic average of the first 50 images belonging to a given noun. The titles are arranged within the poster according to their semantic meaning, using the Wordnet tree to tessalate the 2-D space. As the distance between two tiles relates to their semantic similarity, the relationship between semantic and visual worlds may easily be judged by the viewer. The poster may be viewed at: `http:\\people.csail.mit.edu\torralba\tinyimages`.



a) Person detection

b) Person detection (head size > 20%)

Fig. 12. (a) Recall-precision curves for people detection (not localization) in images drawn randomly from the dataset of 79 million. (b) As per (a) but for the subset of test images where the person's head occupies $> 20\%$ of the image. The left column compares the three different similarity metrics using all 79 millions images. The black line indicates chance-level performance. The graphs on the right compare performance using D_{shift} as a function of dataset size.

for “vise”. In Fig. 9(d) we show the accumulated votes from the neighbors at different levels in the tree, each image voting with unit weight. For clarity, we only show parts of the tree with at least three votes (the full Wordnet tree has $45,815$ non-leaf nodes). The nodes shown in red illustrate the branch with the most votes, which matches the majority of levels in the query image branch (Fig. 9(c)), demonstrating how precise classifications can be made despite significant labeling noise and spurious siblings.

Other work making use of Wordnet includes Hoogs and Collins [21] who use it to assist with image segmentation. While not using Wordnet explicitly, Barnard et al. [2] and Carbonetto et al. [7] learn models using both textual and visual cues.

Using this scheme we now address the task of classifying images of people.

B. *Person detection*

In this experiment, our goal is to label an image as containing a person or not, a task with many applications on the web and elsewhere. A standard approach would be to use a face detector but this has the drawback that the face has to be large enough to be detected, and must generally be facing the camera. While these limitations could be overcome by running multiple detectors, each tuned to different view (e.g. profile faces, head and shoulders, torso), we adopt a different approach.

As many images on the web contain pictures of people, a large fraction (23%) of the 79 million images in our dataset have people in them. Thus for this class we are able to reliably find a highly consistent set of neighbors, as shown in Fig. 10. Note that most of the neighbors match not just the category but also the location and size of the body in the image, which varies considerably in the examples.

To classify an image as containing people or not, we use the scheme introduced in Section V-A, collecting votes from the 80 nearest neighbors. Note that the Wordnet tree allows us make use of hundreds of other words that are also related to “person” (e.g. artist, politician, kid, taxi driver, etc.). To evaluate performance, we used two different sets of test images. The first consisted of a random sampling of images from the dataset. The second consisted of images returned by Altavista using the query “person”.

1) Evaluation using randomly drawn images: 1125 images were randomly drawn from the dataset of 79 million (Fig. 10 shows 6 of them, along some of their sibling set). For evaluation purposes, any people within the 1125 images were manually segmented[12].

Fig. 11 shows the classification performance as the size of the person in the image varies. When the person is large in the image, the performance is significantly better than when it is small. This occurs for two reasons: first, when the person is large, the picture become more constrained, and hence finding good matches becomes easier. Second, the weak labels associated with each image in our dataset typically refer to the largest object in the image.

Fig. 12 shows precision-recall curves as a function of head size, similarly measure and dataset size. As expected, the performance is superior when the person is large in the image and the full 79 million images are used. The D_{shift} similarity measure outperforms both D_{ssd} and D_{warp}.

2) Evaluation using Altavista images: Our approach can also be used to improve the quality of Internet image search engines. We gathered 1018 images from Altavista image search using the keyword “person”. Each image was classified using the approach described in Section V-A. The set of 1018 images was then re-ordered according to the confidence of each classification. Fig. 13(a) shows the initial Altavista ranking while Fig. 13(b) shows the re-ordered set, showing a significant improvement in quality.

To quantify the improvement in performance, the Altavista images were manually annotated with bounding boxes around any people present. Out of the 1018 images, 544 contained people, and of these, 173 images contained people occupying more than 20% of the image.

Fig. 14 shows the precision-recall curves for the people detection task. Fig. 14(a) shows the performance for all Altavista images while Fig. 14(b) shows the performance on the subset where people occupy at least 20% of the image. Note that the raw Altavista performance is the same irrespective of the persons' size (in both plots, by 5% recall the precision is at the level of chance). This illustrates the difference between indexing an image using non visual versus visual cues. Fig. 14 also shows the results obtained when running a frontal face detector[13]. We run the face detector on the original high-resolution images. Note that the performance of our approach working on 32×32 images is comparable to that of the dedicated face detector on

[12] The images and segmentations are available at: `http://labelme.csail.mit.edu/browseLabelMe/static_web_tinyimages_testset.html`

[13] The detector is the OpenCV implementation of the Viola and Jones boosted cascade [26], [43].

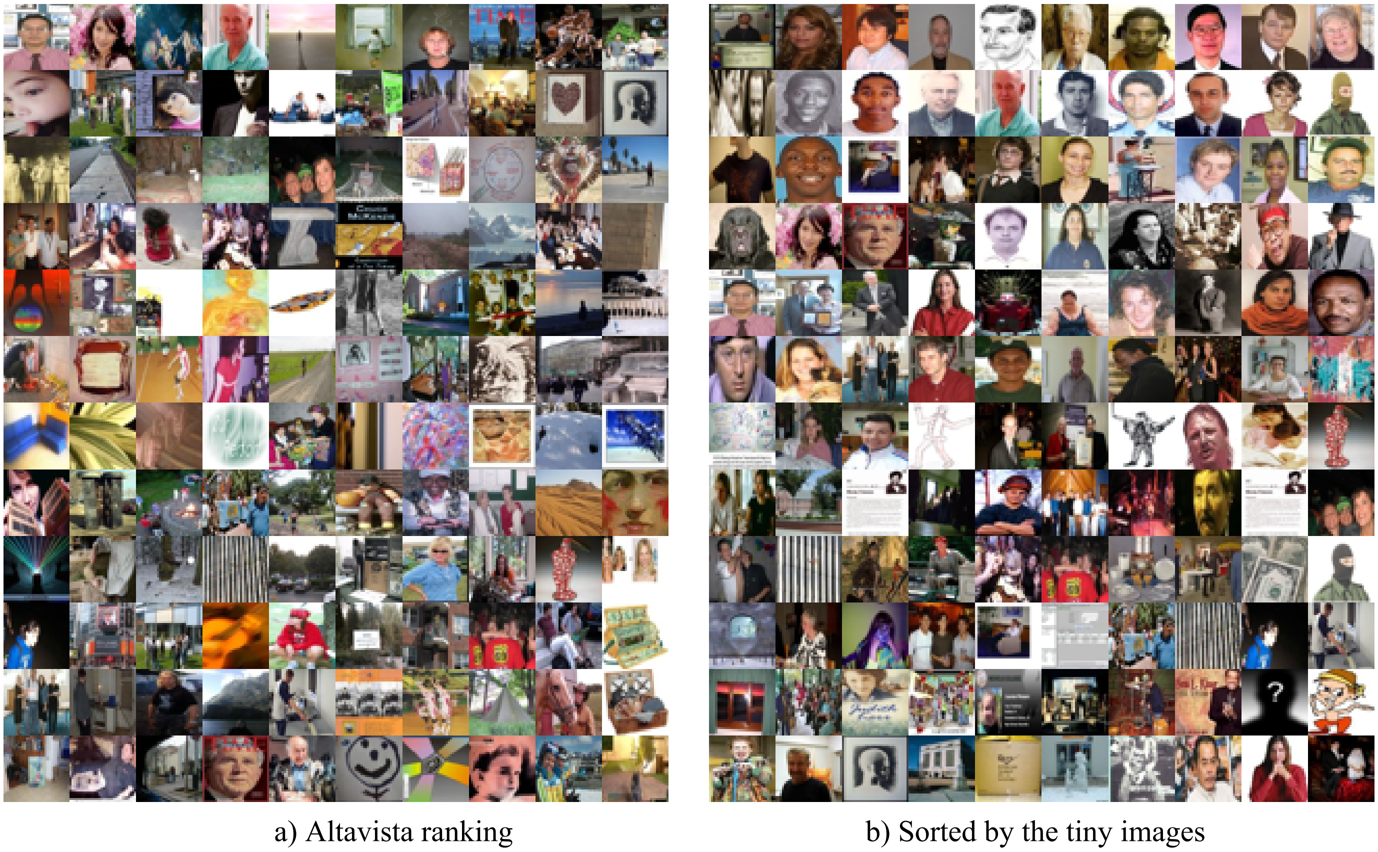


Fig. 13. (a) The first 100 images returned by Altavista when using the query "person" (out of 1018 total). (b) The first 100 images after re-ordering using our Wordnet voting scheme with the 79,000,000 tiny images. This performance improvement is quantified in Fig. 14.

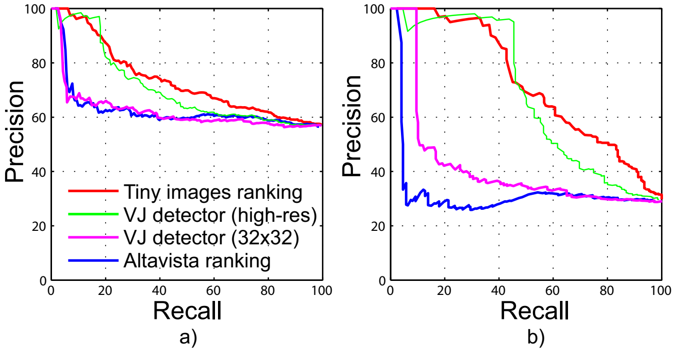


Fig. 14. Evaluation of the results from Fig 13, comparing the performance of the initial Altavista ranking with the re-ordered images using the Wordnet voting scheme and also a Viola & Jones-style frontal face detector. (a) shows the recall-precision curves for all 1018 images gathered from Altavista. (b) shows curves for the subset of 173 images where people occupy at least 20% of the images.

high resolution images. For comparison, Fig. 14 also shows the results obtained when running the face detector on low-resolution images (we downsampled each image so that the smallest axis has 32 pixels, we then upsampled the images again to the original resolution using bicubic interpolation. The upsampling operation was to allow the detector to have enough resolution to be able to scan the image.). The performances of the OpenCV face detector drop dramatically with low-resolution images.

C. *Person localization*

While the previous section was concerned with an object detection task, we now address the more challenging problem of object localization. Even though the tiny image dataset has not been labeled with the location of objects in the images, we can use the weakly labeled (i.e. only a single global label is provided for each image) dataset to localize objects.

Much the recent work in object recognition uses explicit models that labels regions (or pixels) of images as being object/background. In contrast, we use the tiny image dataset to localize without learning an explicit object model. It is important to emphasize that this operation is performed without the need for manual labeling of images: all the information comes from the loose text label associated with each image.

The idea is to extract multiple putative crops of the high resolution query image (Fig. 15a-c). For each crop, we resize it to 32 $\times$ 32 pixels and query the tiny image database to obtain it's siblings set (Fig. 15.d). When a crop contains a person, we expect the sibling set to also contain people. Hence, the most prototypical crops should get have a higher number of votes for the person class. To reduce the number of crops that need to be evaluated, we first segment the image using normalized cuts [9], producing around 10 segments (segmentation is perform on the high resolution image). Then, all possible combinations of contiguous segments are considered, giving a set of putative crops for evaluation. Fig. 15 shows an example of this procedure. Fig. 16 shows the best scoring bounding box for images from the

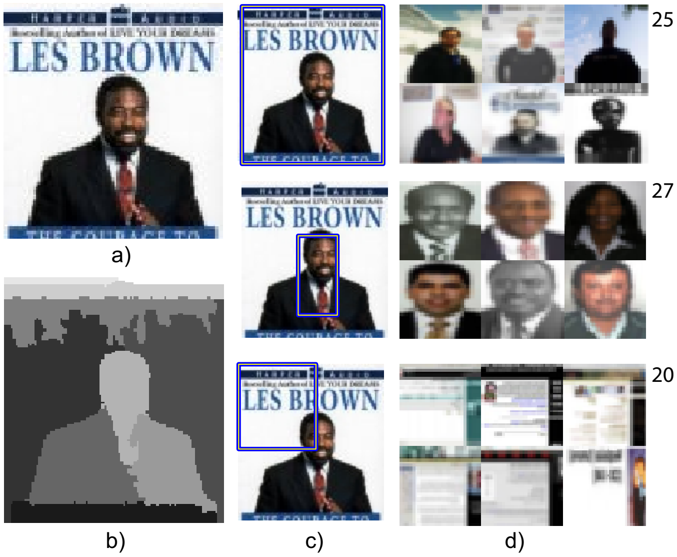


Fig. 15. Localization of people in images. (a) input image, (b) Normalized-cuts segmentation, (c) three examples of candidate crops, (d) the 6 nearest neighbors of each crop in (c), accompanied by the number of votes for the person class obtained using 80 nearest neighbors under similarity measure D_{shift}.

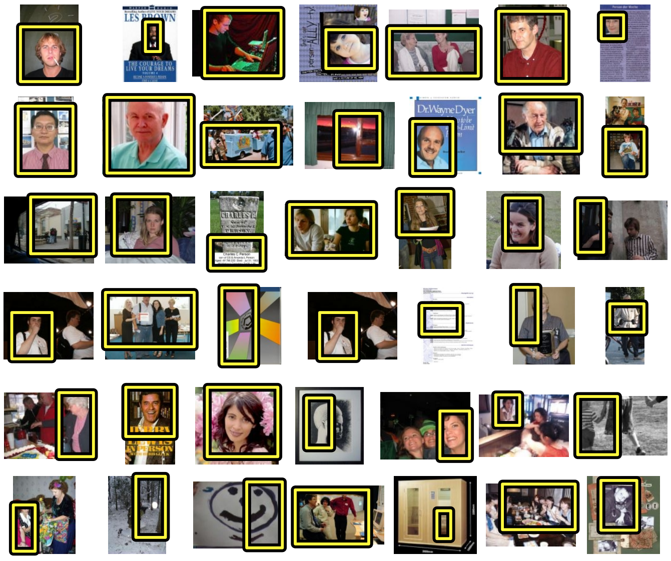


Fig. 16. Localization examples. Images from the 1016 Altavista set overlaid with the crop that gave the highest "person" score. See text for details.

1018 image Altavista test set.

D. Scene recognition

Many web images correspond to full scenes, not individual objects. In this section we use our dataset to classify image between the two; that is to decide that an image is a scene and not a picture of an object. Many nodes in the Wordnet tree refer to scenes and one of the most generic is "location", having children that include "landscape", "workplace", "city" and so on. Using the Wordnet voting scheme of Section V-A, we count the number of votes accumulated at the "location" node of the Wordnet tree to classify a given query. Hopefully, scene images will have a high count, with the reverse being true for object images. With the dataset of 1125 randomly drawn tiny images, of which 185 are scenes, we evaluate the performance at scene versus object classification, the results being shown in Fig. 17. We can also perform classification at a finer semantic level. In Fig. 18, we attempt to classify the 1125 randomly drawn images (containing objects as well as scenes) into "city", "river", "field" and "mountain" by counting the votes at the corresponding node of the Wordnet tree. Scene classification for the 32x32 images performs surprisingly well, exploiting the large, weakly labeled database.

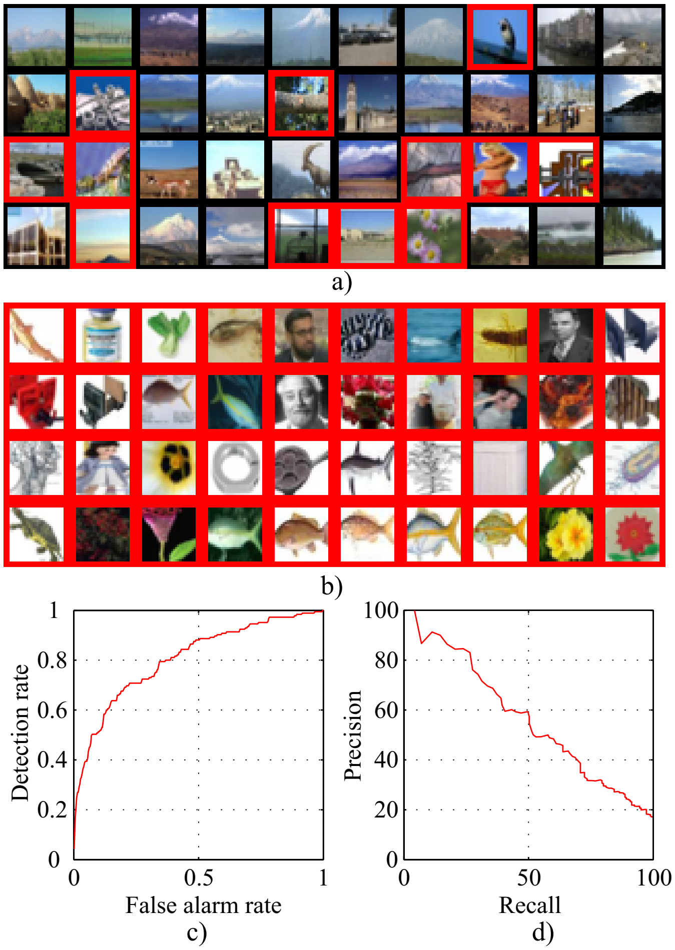


Fig. 17. Classifying between pictures of scenes and objects. (a) Examples of images classified as scenes. The red bounding box denotes a classification error. (b) The set of images having the fewest "location" votes. (c) ROC curve evaluation on test set of 1125 randomly drawn tiny images, of which 185 are scenes. (d) Corresponding precision-recall curve.

E. Automatic image annotation

Here we examine the classification performance at a variety of semantic levels across many different classes, not just people. For evaluation we use the test set of 1125 randomly drawn tiny images, with each image being fully segmented and annotated with the objects and regions that compose each image. To give a distinctive test set, we only use images for which the target object is absent or occupies at least 20% of the image pixels. Using the voting tree described in Section V-A, we classified them using $K = 80$ neighbors at a variety of semantic levels. To simplify the presentation of results, we collapsed the Wordnet tree

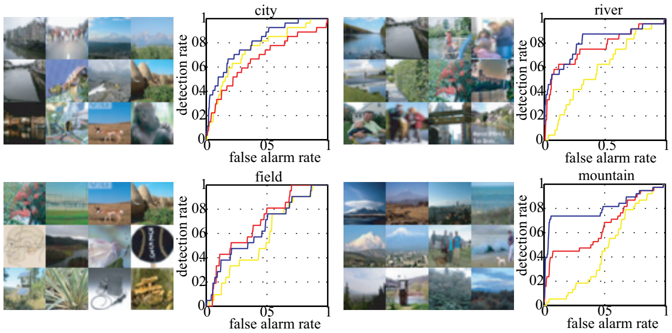


Fig. 18. Scene classification at a finer semantic level than Fig. 17 using the randomly drawn 1125 image test set. Note that the classification is“mountain” vs all classes present in the test set (which includes many kinds of objects), not “mountain” vs “field”, “city”, “river” only. Each quadrant shows some examples of high scoring images for that particular scene category, along with an ROC curve (red = 7,900 image training set; yellow = 790,000 images; blue = 79,000,000 images).

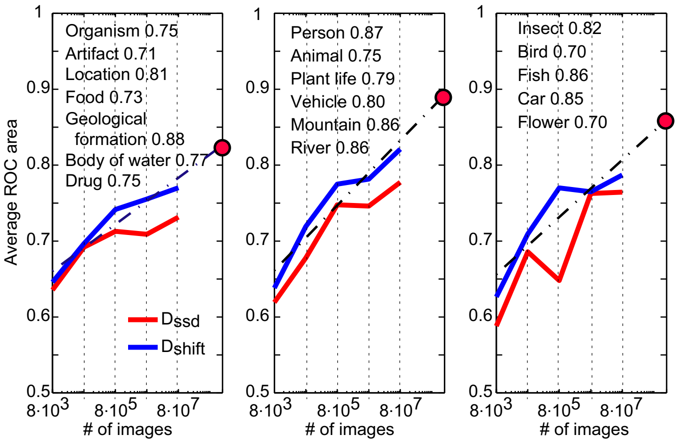


Fig. 19. Classification at multiple semantic levels using 1125 randomly drawn tiny images. Each plot shows a different manually defined semantic level, increasing in selectivity from left to right. The curves represent the average (across words at that level) ROC curve area as function of number of images in the dataset (red=D_{ssd}, blue=D_{shift}). Words within each of the semantic levels are shown in each subplot, accompanied by the ROC curve area when using the full dataset. The red dot shows the expected performance if all images in Google image search were used (∼2 billion), extrapolating linearly.

by hand (which had 19 levels) down to 3 levels corresponding to one very high level (“organism”, “artifact”), an intermediate level (“person”, “vehicle”, “animal”) and a level typical of existing datasets (“fish”, “bird”, “car”).

In Fig. 19 we show the average ROC curve area (across words at that level) at each of the three semantic levels for D_{ssd} and D_{shift} as the number of images in the dataset is varied. Note that (i) the classification performance increases as the number of images increases; (ii) D_{shift} outperforms D_{ssd}; (iii) the performance drops off as the classes become more specific.

By way of illustrating the quality of the recognition achieved by using the 79 million weakly labeled images, we show in Fig. 20, for categories at three semantic levels, the images that were more confidently assigned to each class. Note that despite the simplicity of the matching procedure presented here, the recognition performance achieves reasonable levels even for relatively fine levels of categorization.

VI. Other Applications

In this section we discuss other applications, beyond recognition, that rely on a dense sampling of the manifold of natural images. We present two applications: (i) image colorization of gray scale images; (ii) detecting image orientation.

A. *Image colorization*

The goal of image colorization is to recover the missing color information from a gray scale image. This task is generally solved by having a user specify colors that the different image regions will have and then using a diffusion process to propagate the color to the rest of the image. Given a gray scale query image, we propose to use the sibling set to define a distribution of possible colors for it, with no human labeling. The assumption is that images in a neighborhood contain similar objects arranged in similar locations, thus the colors should be exchangeable among the images in the sibling set. Clearly, this will only work when the neighborhood of the query image is densely populated, hence large amounts of images will be required.

Fig. 21 shows the various stages of our approach. We start with a gray scale image (first row). We search in the tiny image database for similar images using only gray scale information (second row). Then, for each of the retrieved siblings, we download the original high resolution color image (third row). The idea is to use the colors from the sibling images to colorize the gray scale input image. One possible approach is to compute the arithmetic average of the color sibling images (as shown in the fourth row). The color channels a & b (from the Lab transformed image) for each pixel from the average sibling image are copied to the input image, so colorizing it. When the siblings are very similar to one another, the average appears sharp and the colorized image is compelling (fifth row). Alternatively, we can copy the color channels from individual siblings to propose multiple plausible colorizations of the input image (sixth row). The user can then select the best one. While simple, our approach has its limitations: by directly copying the color information across, the edges in the input image are not respected. It would be possible to improve the quality of the results by using a method such as Levin et al. [24] which would take color cues from the sibling images and propagate within the input image in an edge-aware manner.

The last column of figure 21 illustrates the sensibility of this approach to the manifold of natural images. In this example, the input is a picture upside-down. As this is not a typical picture orientation, the image is slightly outside of the manifold of typical natural images and the retrieved sibling set is not consistent anymore. The average sibling image is not as sharp as the one obtained when the picture was in the correct orientation. The procedure does not provide plausible colorizations. This opens the door to automatically predict what is the correct orientation of a picture as discussed in the next section.

B. *Detecting image orientation*

Although it is easy for us to tell when a picture it is upside down, currently there is no satisfactory way of doing thus automatically. We present a simple approach using our large dataset of tiny images. We argue is that when an image has the wrong orientation, it becomes harder to find good neighbors since it starts to move away from the manifold defined by the set of

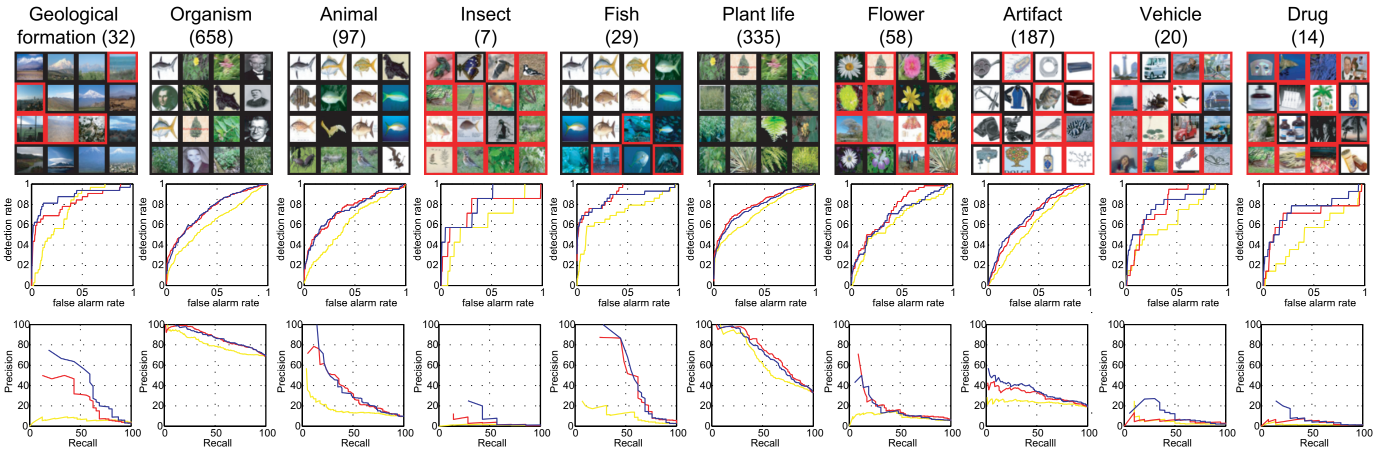


Fig. 20. Test images assigned to words at each semantic level. The images are ordered by voting confidence. The number indicates the total number of positive examples in the test set out of the 1148 images. The color of the bounding box indicates if the image was correctly assigned (black) or not (red). The middle row shows the ROC curves for three dataset sizes (red = 7,900 image training set; yellow = 790,000 images; blue = 79,000,000 images). The bottom row shows the corresponding precision-recall graphs.

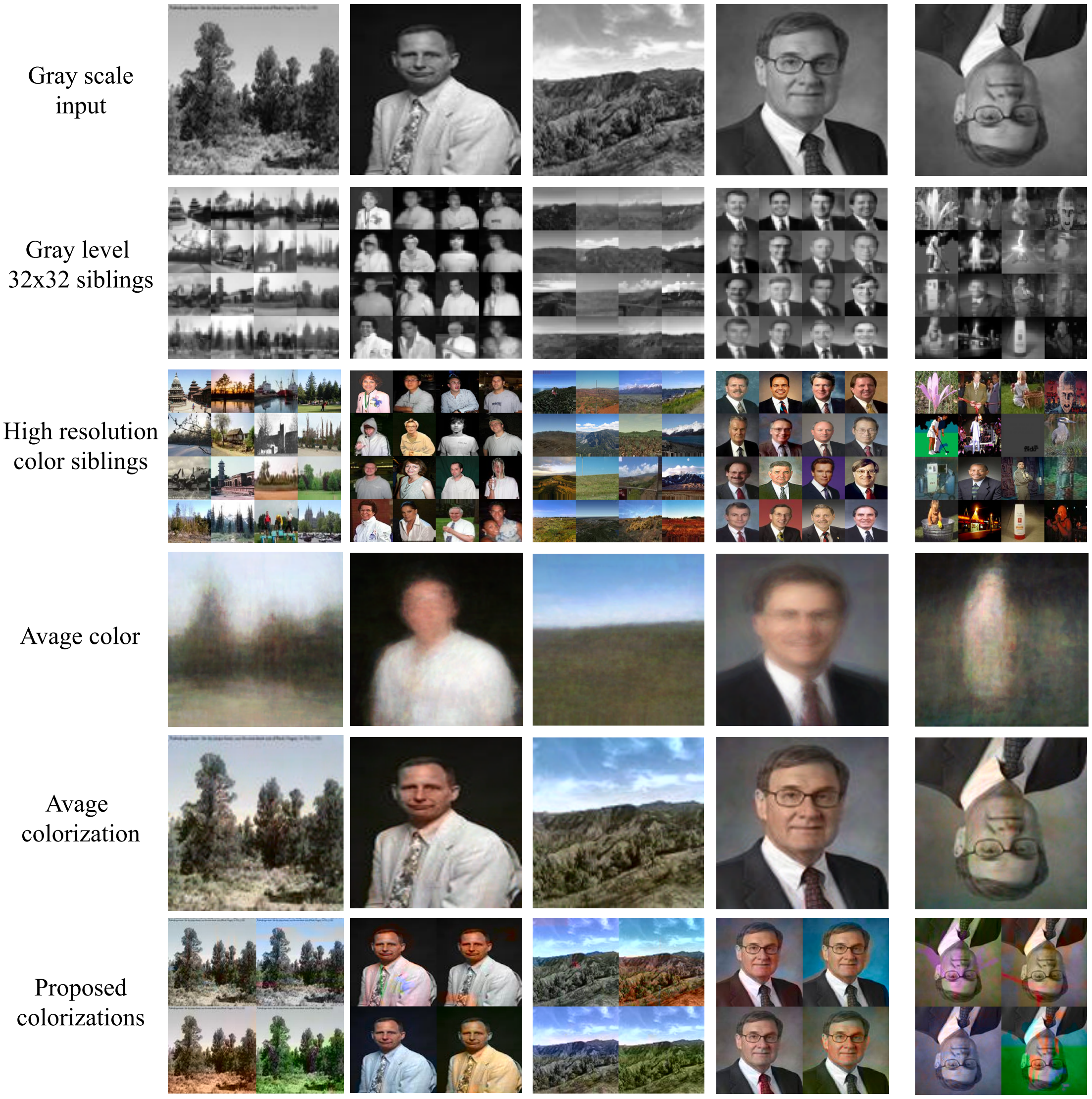


Fig. 21. Automatic image colorization. From top to bottom, first row, gray scale input image, second row, 32×32 gray scale siblings, third row, corresponding high resolution color siblings, fourth row, average of the color siblings, fifth row, input image with color from the average, sixth row, candidate colorizations by taking the color information from four different siblings.

correctly oriented (i.e., camera parallel to the horizon line) natural images. Thus the correct orientation may be found by selecting the image orientation that maximizes the typicality of the image. Fig. 22 shows three examples of images being evaluated at four possible orientations. The number on top of each image is the average correlation $(1-D_{warp}/2)$ to the 50 closest neighbors. The red boundary denotes the preferred orientation, namely the one with the highest average correlation. Fig. 23 shows a quantitative evaluation using the test set of 1125 images randomly drawn from the tiny images (as introduced in Section V-B.1) being classified into one of four possible orientations. Many images in this test set are ambiguous in terms of orientation, making them hard to classify correctly (see Fig. 23(a)). Thus, if only those images that are classified with high confidence are considered, the performance is much improved (see Fig. 23(b)).

Our procedure differs dramatically from that of Vailaya et al. [42] who vector-quantize image patches using a pre-built codebook and model the resulting representation using parametric models to predict image orientation. Instead, our non-parametric approach relies entirely on a large dataset to give us an accurate measure of the distance from the manifold of natural images.

VII. Conclusions

This paper makes five important contributions:

1) Compelling psychophysical experiments showing that 32 × 32 is the minimum color image resolution at which object and scene recognition can reliably be performed.
2) The compilation of a dataset of 79 million 32 × 32 color images, each with a weak text label and link to the original high-resolution image, which is available for download.
3) The characterization of the manifold of 32 × 32 images, showing that Internet sized datasets (10^8–10^9) yield a reasonable density over the manifold of natural images, at least from a categorization perspective.
4) The demonstration that simple non-parametric methods, in conjunction with the tiny image dataset, can give reasonable performance on object recognition tasks. For classes which are richly represented, such as people, the performance is comparable to leading class-specific detectors.
5) The novel application of the tiny image dataset to a variety of other problems in computer vision, such as image colorization and orientation determination.

Although the paper explores many topics, it has one key theme: that of using non-parametric methods in conjunction with very large collections of images to tackle object and scene recognition. Previous usage of non-parametric approaches in recognition have been confined to more limited domains (e.g. pose recognition [36]) compared with the more general problems tackled in this paper, the limiting factor being the need for very large amounts of training data. The results obtained using our tiny image dataset are an encouraging sign that the data requirements may not be insurmountable. Indeed, search engines such as Google index another 1–2 orders of magnitude more images, which could yeild a significant improvement in performance.

While the Internet offers a plentiful supply of visual data, there are drawbacks to using it as a source. First, the high noise level in image labellings make it difficult to directly train models without some method for ignoring the numerous outliers. Although the Wordnet scheme we propose gives some benefit, the results of all our experiments would undoubtedly be much

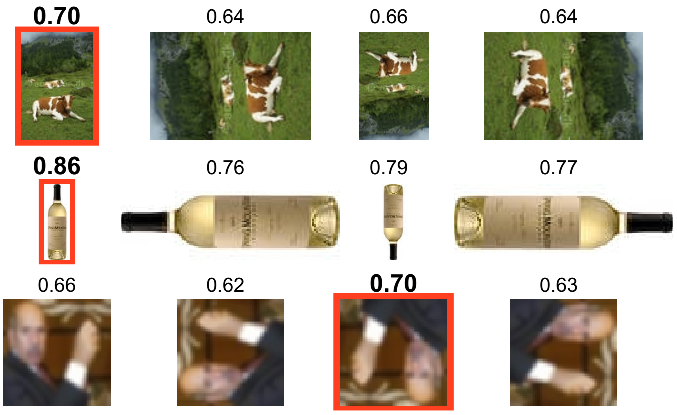


Fig. 22. Automatic image orientation determination. For each of the three example images we consider four possible orientations. The number on top of each image is the average correlation $(1 - D_{\text{warp}}/2)$ to the 50 closest neighbors. The red boundary denotes the preferred orientation. The last example is an error.

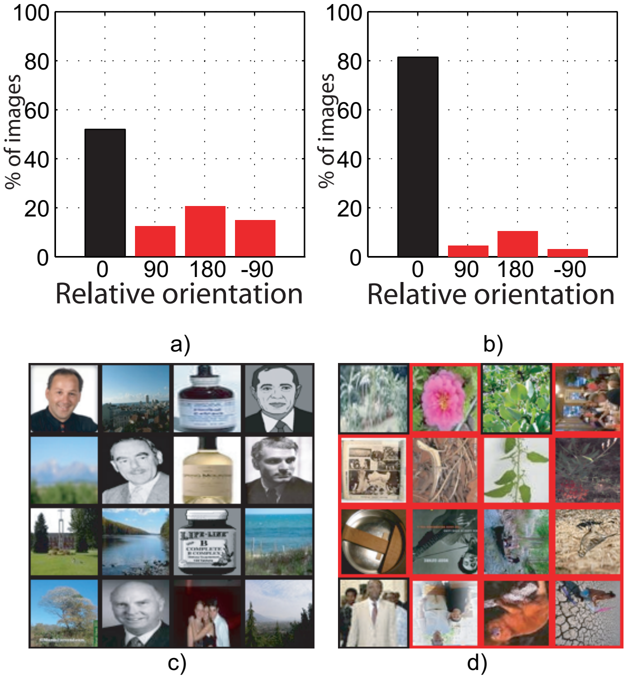


Fig. 23. (a) Distribution of assigned orientations, relative to the correct one, in the set of 1125 randomly drawn images from the tiny image database. 52% of the images were assigned the correct orientation. Most of the errors correspond to selecting an upside down version of the right image orientation. (b) Distribution of assigned image orientations for the 25% of the test set with highest confidence. In this portion of the test set, 81.9% of the images are assigned the correct image orientation. (c) Examples of the 16 images with highest classification confidence. (d) Examples of the 16 images with lowest classification confidence. Images with a red boundary are errors. Note that many of these images have no distinct orientation so are hard to classify correctly.

improved if there were less labeling noise. Second, the images themselves have peculiar statistics, different to other sources of images (e.g. television or the view through our own eyes) in terms of both their content and their class-distribution. For example, many images contain centered views of objects. Also, there are a disproportionate number of images containing people on the Internet. Although this facilitates person detection, many classes are not well represented in our tiny image dataset. It would therefore be interesting to explore large collections of images from alternative sources, such as video.

The technical methods used in this paper are simple and more complex ones are likely to improve performance in a number of areas. Better similarity metrics might give a significant increase in the effective size of the dataset. Machine learning techniques could be effective in reducing labeling noise, which in turn would improve performance. Also, efficient search methods would give real-time recognition performance.

In summary, all methods in object recognition have two components: the model and the data. The vast majority of the effort in recent years has gone into the modeling part – seeking to develop suitable parametric representations for recognition. In contrast, this paper moves in the opposite direction, exploring how the data itself can help to solve them problem. We feel the results in this paper warrant further exploration in this direction.

References

[1] T. Bachmann. Identification of spatially queatized tachistoscopic images of faces: How many pixels does it take to carry identity? *European Journal of Cognitive Psychology*, 3:85–103, 1991.

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, 2003.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Info. Proc. Systems*, pages 831–837, 2000.

[4] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. Technical report, UC Berkeley, 2004.

[5] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, volume 1, pages 26–33, June 2005.

[6] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, volume 2, pages 1463–1470, 2006.

[7] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *ECCV*, 2004.

[8] D. M. Chandler and D. J. Field. Estimates of the information content and dimensionality of natural scenes from proximity distributions. *JOSA*, 2006.

[9] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, 2005.

[10] M. Everingham, A. Zisserman, C.K.I. Williams, and L. Van Gool. The PASCAL visual object classes challenge 2006 (voc 2006) results. Technical report, University of Oxford, September 2006. Available from http://www.pascal-network.org/challenges/VOC/voc2006/.

[11] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene? *Journal of Vision*, 7(1):1–29, 2007.

[12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[13] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. Bradford Books, 1998.

[14] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proceedings of the 10th International Conference on Computer Vision, Beijing, China*, volume 2, pages 1816–1823, October 2005.

[15] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, pages 242–256, May 2004.

[16] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos at a glance. In *Proc. Intl. Conf. Pattern Recognition*, volume 1, pages 459–464, 1994.

[17] K. Grauman and T. Darrell. Pyramid match hashing: Sub-linear time indexing over partial correspondences. In *CVPR*, 2007.

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report UCB/CSD-04-1366, Caltech, 2007.

[19] L. D. Harmon and B. Julesz. Masking in visual recognition: Effects of two-dimensional noise. *Science*, 180:1194–1197, 1973.

[20] J. Hays and A. A. Efros. Scene completion using millions of photographs. *SIGGRPAH, ACM Transactions on Graphics*, 26, 2007.

[21] A. Hoogs and R. Collins. Object boundary detection in images using a semantic ontology. In *AAAI*, 2006.

[22] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[23] Ann B. Lee, Kim S. Pedersen, and David Mumford. The nonlinear statistics of high-contrast patches in natural images. *Int. J. Comput. Vision*, 54(1-3):83–103, 2003.

[24] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *SIGGRAPH, ACM Transactions on Graphics*, 2004.

[25] J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic object picture collection via incremental model learning. In *CVPR*, 2007.

[26] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In *DAGM 25th Pattern Recognition Symposium*, 2003.

[27] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision ICCV, Corfu*, pages 1150–1157, 1999.

[28] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging understanding workshop*, pages 121–130, 1981.

[29] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[30] A. Oliva. Gist of the scene. In *Neurobiology of Attention, L. Itti, G. Rees & J. K. Tsotsos (Eds.)*, pages 251–256, 2005.

[31] A. Oliva and P.G. Schyns. Diagnostic colors mediate scene recognition. *Cognitive Psychology*, 41:176–210, 1976.

[32] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Intl. J. Computer Vision*, 42(3):145–175, 2001.

[33] M.C. Potter. Short-term conceptual memory for pictures. *Journal of Experimental Psychology: Human Learning and Memory*, 2:509–522, 1976.

[34] L. Renninger and J. Malik. When is scene recognition just texture recognition? *Vision Research*, 44:2301–231, 2004.

[35] B. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman. Object recognition by scene alignment. In *Advances in Neural Info. Proc. Systems*, page To appear, 2007.

[36] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *IEEE Intl. Conf. on Computer Vision*, 2003.

[37] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):137–154, 2006.

[38] A. Torralba. Contextual priming for object detection. *Intl. J. Computer Vision*, 53(2):153–167, 2003.

[39] A. Torralba, R. Fergus, and W.T. Freeman. Tiny images. Technical Report MIT-CSAIL-TR-2007-024, CSAIL, MIT, 2007.

[40] A. Torralba, K. Murphy, W. Freeman, and M. Rubin. Context-based vision system for place and object recognition. In *Intl. Conf. Computer Vision*, 2003.

[41] A. Torralba and A. Oliva. Depth estimation from image structure. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1225, 2002.

[42] A. Vailaya, H. Zhang, C. Yang, F. Liu, and A. Jain. Automatic image orientation detection. *IEEE Trans. on Image Processing*, 11(7), July 2002.

[43] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple classifiers. In *CVPR*, 2001.

[44] J. Wolfe. Visual memory: What do you know about what you saw? *Current Biology*, 8:R303–R304, 1998.