Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2007-024 April 23, 2007

Tiny images

Antonio Torralba, Rob Fergus, and William T. Freeman

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Tiny images

Antonio Torralba Rob Fergus William T. Freeman
CSAIL, Massachusetts Institute of Technology,
32 Vassar St., Cambridge, MA 02139, USA
{torralba,fergus,billf}@csail.mit.edu

Abstract

The human visual system is remarkably tolerant to degradations in image resolution: in a scene recognition task, human performance is similar whether 32×32 *color images or multi-mega pixel images are used. With small images, even object recognition and segmentation is performed robustly by the visual system, despite the object being unrecognizable in isolation. Motivated by these observations, we explore the space of* 32×32 *images using a database of* 10^8 32×32 *color images gathered from the Internet using image search engines. Each image is loosely labeled with one of the* $70,399$ *non-abstract nouns in English, as listed in the Wordnet lexical database. Hence the image database represents a dense sampling of all object categories and scenes. With this dataset, we use nearest neighbor methods to perform object recognition across the* 10^8 *images.*

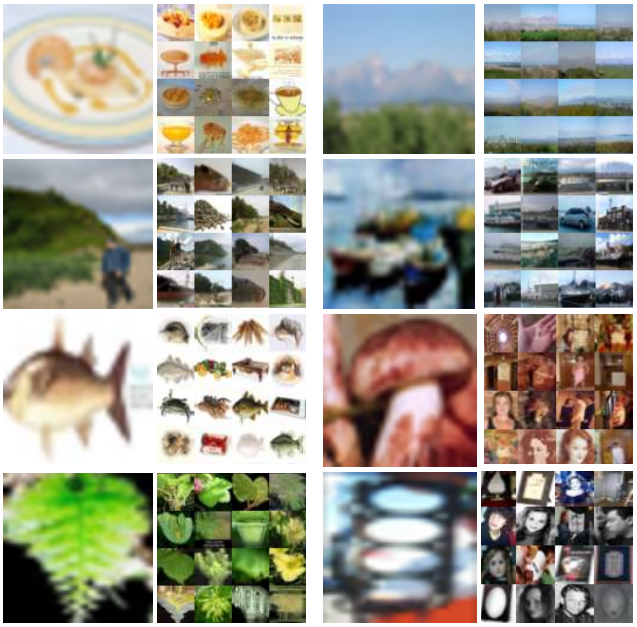

Figure 1. 1st & 3rd columns: Eight 32×32 resolution color images. Despite their low resolution, it is still possible to recognize most of the objects and scenes. These are samples from a large dataset of 10^8 32×32 images we collected from the web which spans all visual object classes. 2nd & 4th columns: 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.

1 Introduction

When we look 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 seem to already contain most of the relevant information needed to support reliable recognition.

Motivated by our ability to perform object and scene recognition using very small images, in this paper we explore a number of fundamental questions: (i) what is the smallest image dimensionality that suffices? (ii) how many different tiny images are there? (iii) how much data do we need to viably perform recognition with nearest neighbor approaches?

Currently, most successful computer vision approaches to scene and object recognition rely on extracting textural cues, edge fragments, or patches from the image. These methods require high-resolution images since only they can provide the rich set of features required by the algorithms. Low resolution images, by contrast, provide a nearly independent source of information to that presently extracted from high resolution images by feature detectors and the like. Hence any method successful in the low-resolution domain can augment existing methods suitable for high-resolution images.

Another benefit of working with tiny images is that it becomes practical to store and manipulate datasets orders of magnitude bigger than those typically used in computer vision. Correspondingly, we introduce a dataset of 70 million unique 32×32 color images gathered from the Internet. Each images is loosely labelled with one of $70,399$ 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.

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?" tools which corrects errors in search queries, not through a complex parsing of the query but by memorizing billions of correct query strings and suggesting the closest to the users query. We explore a visual analogy to this tool using our dataset and nearest-neighbor matching schemes.

Nearest-neighbor methods have previously been used in a variety of computer vision problems, primarily for interest point matching [3, 12, 17]. It has also been used for global image matching, albeit in more restrictive domains such as pose estimation [24].

The paper is divided into three parts. In Section 2 we investigate the performance of human recognition on tiny images, establishing the minimal resolution required for robust scene and object recognition. In Sections 3 and 4 we introduce our dataset of 70 million images and explore the manifold of images within it. 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, despite the labelling noise in the dataset.

2 Human recognition of low-resolution images

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, 14] have shown that only 16×16 pixels are needed for robust face recognition. This remarkable performance is also found in a scene recognition task [20]. However, there are no studies that have explored the minimal image resolution required to perform visual tasks such as generic object recognition, segmentation, and scene recognition with many categories. In computer vision, existing work on low-resolution images relies on motion cues [7].

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). 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. [19, 22, 23]. Here, we are interested in characterizing the amount of information available in the image as a function of the image resolution (there is no constraint on presentation time). We start with a scene recognition task.

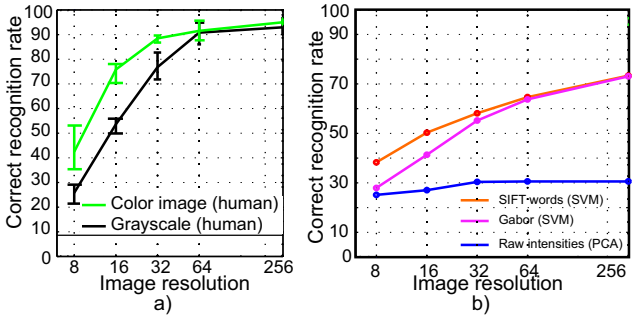

Figure 2. a) Human performance on scene recognition as a function of resolution. The green and black curves shows the performance on color and grayscale images respectively. For color 32×32 images the performance only drops by 7% relative to full resolution, despite having 1/64th of the pixels. (b) Computer vision algorithms applied to the same data as (a). A baseline algorithm (blue) and state-of-the-art algorithms [16, 21].

2.1 Scene recognition

In cognitive psychology, the *gist* of the scene [19, 25] 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 that provides sufficient information for scene recognition and context for object detection. In this section, we show that this low dimensional representation can rely on very low-resolution information and, therefore, can be computed very efficiently.

We evaluate the scene recognition performance of both humans and existing computer vision algorithms[9, 16, 21] as the image resolution is decreased. The test set of 15 scenes was taken from [16]. Fig. 2(a) shows human performance on this task when presented with grayscale and color images[2] 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. Interestingly, when color and grayscale results are plotted against image dimensionality (number of pixels $\times$ color bands) the curves for both color and grayscale images overlap (not shown). Therefore, humans need around 3072 dimensions of either color or grayscale data to perform this task. Fig. 2(b) compares human performance (on grayscale data) with state of the art computer vision algorithms as a function of image

[1] 32×32 is very very small. For reference, typical thumbnail sizes are: Google images (130x100), Flikr (180x150), default Windows thumbnails (90x90).

[2] 100% recognition rate can not be achieved in this dataset as there is no perfect separation between the 15 categories

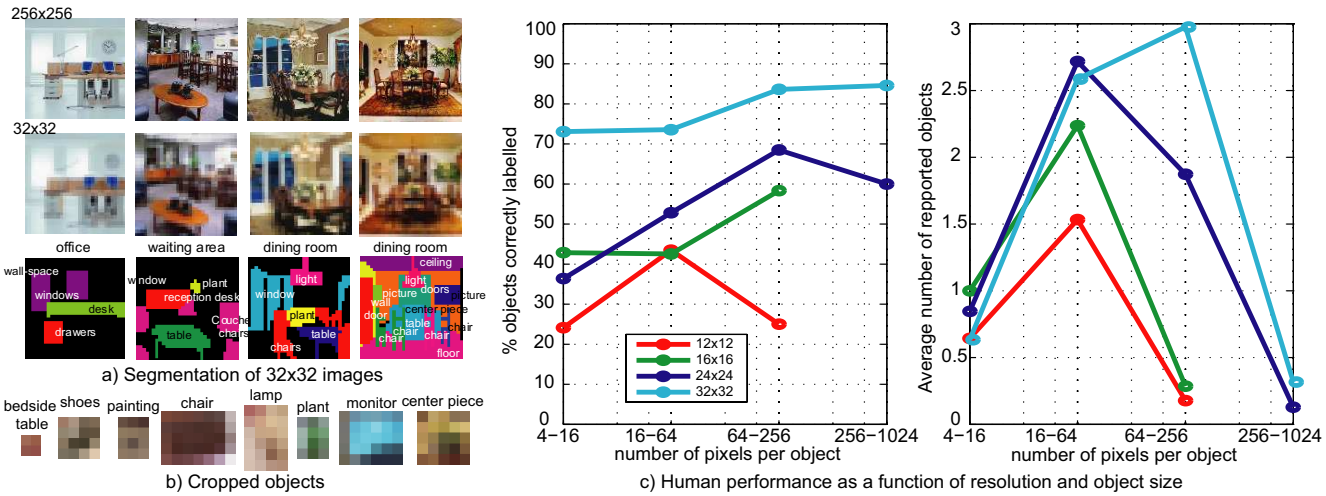

Figure 3. Human segmentation of tiny images. (a) Humans can correctly recognize and segment objects at very low resolutions, even when the objects in isolation can not be recognized (b). c) Summary of human performances as a function of image resolution and object size. Humans analyze images quite well at 32×32 resolution.

resolution. The algorithms used for scene recognition are: 1) PCA on raw intensities, 2) a SVM classifier on a vector of Gabor filter outputs[21], and 3) a descriptor built using histograms of quantized SIFT features ([16]). We used 100 images from each class for training as in [16]. Raw intensities perform very poorly. The best algorithms are (magenta) Gabor descriptors[21] with a SVM using a Gaussian kernel and (orange) the SIFT histograms[16][3] There is not a significant difference in performance between the two. All the algorithms show similar trends, with performances at 256^2 pixels still below human performance at 32^2.

2.2 Object recognition

Recently, the PASCAL object recognition challenge evaluated a large number of algorithms in a detection task for several object categories [8]. Fig. 4 shows the performances (ROC) of the best performing algorithms in the competition in the car classification task (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 at very low-resolutions. Human participants were shown color images from the test set scaled to have 32 pixels on the smallest axis. Fig. 4 shows some examples of tiny PASCAL images. Each participant classified between 200 and 400 images selected randomly. Fig. 4 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.

[3]SIFT descriptors have an image support of 16×16 pixels. Therefore, when working at low resolutions it was necessary to upsample the images. The best performances were obtained when the images were upsampled to 256^2.

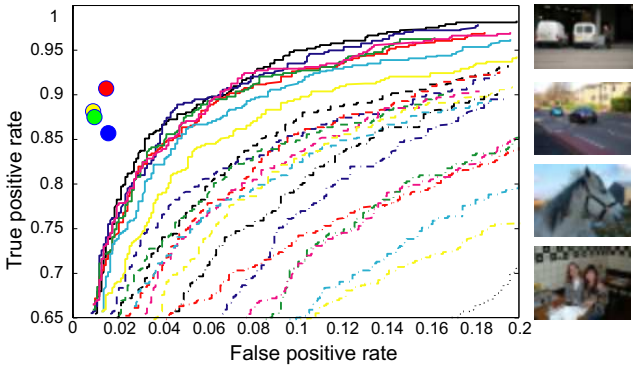

Figure 4. Car detection task on the PASCAL 2006 test dataset. The colored dots show the performance of four human subjects classifying tiny versions of the test data. The ROC curves of the best vision algorithms (running on full resolution images) are shown for comparison. All lie below the performance of humans on the tiny images, which rely on none of the high-resolution cues exploited by the computer vision algorithms.

2.3 Object segmentation

A more challenging task is that of object segmentation. Here, participants are shown color images at different resolutions (12^2, 16^2, 24^2, and 32^2) and their task is to segment and categorize as many objects as they can. Fig. 3(a) shows some of the manually segmented images at 32^2. It is important to note that taking objects out of their context drastically reduces recognition rate. Fig. 3(b) shows crops of some of the smallest objects correctly recognized. The resolution is so low that recognition of these objects is almost entirely based on context. Fig. 3(c) shows human performance (evaluation is done by a referee that sees the original high resolution image and the label assigned by the participant. The referee does not know at which resolution the image was presented). The horizontal axis corresponds to the number of pixels occupied by the object in the image. The

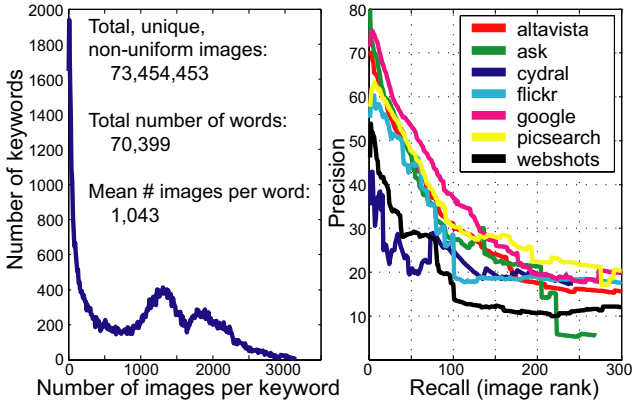

Figure 5. Statistics of the tiny images database. a) A histogram of images per keyword collected. Around 10% of keywords have very few images. b) Performance of the various engines (evaluated on hand-labeled ground truth). Google and Altavista are the best performing and Cydral and Flickr the worst.

two plots show the recognition rate (% objects correctly labelled) and the number of objects reported for each object size. Each curve corresponds to a different image resolution. At 32^2 participants report around 8 objects per image and the correct recognition rate is around 80%. Clearly, sufficient information remains for reliable segmentation.

Of course, not all visual tasks can be solved using such low resolution images, the experiments in this section having focused only on recognition tasks. However, we argue 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 performance drop.

3 A large dataset of 32x32 images

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 [13]. 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 10^8 such images, made possible by the minimal storage requirements for each image.

3.1 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, $75,378$ of them in total. Note that 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. Fig. 5(a) shows a histogram of the number of images per class.

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 $75,378$ nouns. Running over 6 months, this method gathered $95,707,423$ images in total. Once intra-word duplicates and uniform images (images with zero variance) are removed, this number is reduced to $73,454,453$ images from $70,399$ words (around 10% of the 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[6]. The dataset fits onto a single hard disk, occupying 600Gb in total.

3.2 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. In Fig. 5(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 [5] 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.* [11, 10] have shown the use of a variety of approaches for improving Internet image search engines. 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 labelling noise.

[4] Wordnet [26] 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 category-level classes and more specific objects such as US presidents, astronomical objects and Abyssinian cats.

[6] We also stored a version maintaining the original aspect ratio (the minimum dimension was set at 32 pixels) and a link to the original thumbnail and high resolution URL.

4 The manifold of natural images

Using the dataset we are able to explore the manifold of natural images[7]. Despite 32×32 being very low resolution, each image lives in a space of 3072 dimensions. This is still a huge space - if each dimension has 8 bits, there are a total of 10^{7400} possible images. 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. To measure this fraction we must first define an appropriate distance metric to use in the 3072 dimensional space.

We introduce three different distance measures between a pair of images i_1 and i_2. We assume that all images have already been normalized to zero mean, unit variance.

- Sum of squared differences (SSD) between the normalized images (across all three color channels). Note that $D_1 = 2(1-\rho)$, ρ being the normalized correlation.

$$D_1 = \sum_{x,y,c} (i_1(x,y,c) - i_2(x,y,c))^2$$

- Warping. We optimize each image by transforming i_2 (horizontal mirror; translations and scalings up to 10 pixel shifts) to give the minimum SSD. The transformation parameters θ are optimized by gradient descent.

$$D_2 = \min_{\theta} \sum_{x,y,c} (i_1(x,y,c) - T_\theta[i_2(x,y,c)])^2$$

- Pixel shifting. We allow for additional distortion in the images by shifting every pixel individually within a 5 by 5 window to give minimum SSD ($w = 2$). We assume that i_2 has already been warped: $\hat{i}_2 = T_\theta[i_2]$. The minimum can be found by exhaustive evaluation of all shifts, only possible due to the low resolution of the images.

$$D_3 = \min_{|Dx,y| \le w} \sum_{x,y,c} (i_1(x,y,c) - \hat{i}_2(x+D_x, y+D_y, c))^2$$

Computing distances to $70,000,000$ images is computationally expensive. To improve speed, we index the images using the first 20 principal components of the $70,000,000$ images. With only 20 components, all 3 metrics are equivalent and the entire index structure can be held in memory. Using exhaustive search we find the closest 4000 images in 30 seconds[8] per image. The distances D_1, D_2, D_3 to these 4000 neighbors are then computed, D_1 taking negligible time while D_2 and D_3 take a minute or so. Fig. 6 shows a pair of images being matched using the 3 metrics. Fig. 1 shows examples of query images and sets of neighboring images from our dataset found using D_3.

Inspired by studies on the statistics of image patches [6], we use our dataset to explore the density of images using D_1. Fig. 7 shows several plots measuring various properties as the size of the dataset is increased. In Fig. 7(a), we show the probability that the nearest neighbor has a normalized correlation exceeding a certain value. Fig. 7(b) shows a vertical section through Fig. 7(a) at 0.8 and 0.9 as the number of images grows logarithmically. Fig. 7(c) shows the probability of the matched image being a duplicate as a function of D_1. While we remove duplicates *within* each word, it is not trivial to remove them *between* words.

Figure 6. Image matching using distance metrics D_1, D_2 and D_3. For D_2 and D_3 we show the closest manipulated image to the target.

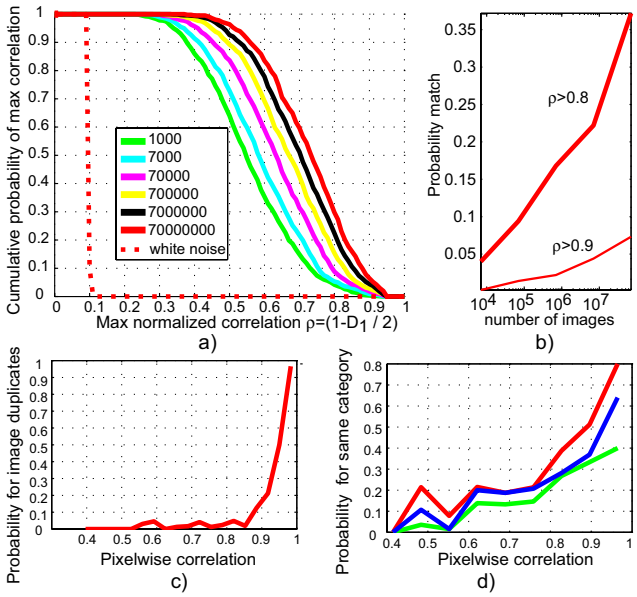

Figure 7. Exploring the dataset using D_1. (a) Cumulative probability for the correlation with the nearest neighbor. (b) Cross-section of figure (a) plots the probability of finding a neighbor with correlation > 0.9 as a function of dataset size. (c) Probability that two images are duplicates as a function of pixelwise correlation. (d) Probability that two images belong to the same category as a function of pixelwise correlation (duplicate images are removed). Each curve represents a different human labeler.

[7] Although our dataset is 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.

[8] Undoubtedly, if efficient data structures such as a kd-tree were used, the matching would be significantly faster. Nister and Stewenius [18] used related methods to index over 1 million images in $\sim$ 1sec.

In Fig. 7(d) we explore how the plots shown in Fig. 7(a) & (b) relate to recognition performance. Three human subjects labelled pairs of images as belonging to the same visual class or not. As the normalized correlation exceeds 0.8, the probability of belonging to the same class grows rapidly. From Fig. 7(b) we see that a quarter of the images in the dataset are expected to have a neighbor with correlation > 0.8. Hence a simple nearest-neighbor approach might be effective with our size of dataset.

5 Recognition

We now attempt to recognize objects and scenes in our dataset. While a variety of existing computer vision algorithms 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_1, D_2 or D_3. 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 query image with some semantic connection to it.

Given the large number of classes in our dataset $(70,399)$ and their highly specific nature, it is not practical or desirable to try and classify each of the classes separately. Instead we make use of Wordnet [26] which provides semantic hierarchy (hypernyms) for each noun. Using this hierarchy, we can perform classification at a variety of different semantic levels, thus instead of trying to recognize the noun "yellowfin tuna" we can also perform recognition at the level of "tuna" or "fish" or "animal". Other work making use of Wordnet includes Hoogs and Collins [15] who use it to assist with image segmentation. Barnard *et al.* [2] showed how to learn simultaneously the visual and text tags of images etc.

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

5.1 Classification using Wordnet voting

Wordnet provides semantic relationships between the $70,399$ nouns for which we have collected images. We decompose the graph-structured relationships into a tree by taking the most common meaning of each word. This tree is then used to accumulate votes from the set of neighbors found for a given query image. Each neighbor has its own branch within the tree for which it votes. By accumulating these branches the query image may be classified at a variety of levels within the tree.

In Fig. 8(a) we show a query image of a vise from our test set. In Fig. 8(b) we show a selection from the $K = 80$ nearest neighbors using D_3 over the 70 million images. In Fig. 8(c) we show the Wordnet branch for "vise". In Fig. 8(d) we show the accumulated votes from the neighbors

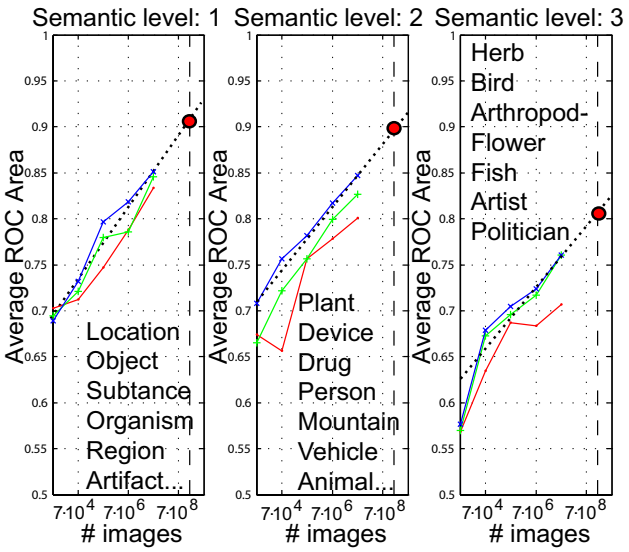

Figure 9. Average ROC curve area at different semantic levels as a function of number of images in the dataset, for D_1 (red), D_2 (green) and D_3 (blue). Words within each of the semantic levels are shown in each subplot. The red dot shows the expected performance if all images in Google image search were used (~ 2 billion), extrapolating linearly.

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 query image branch (Fig. 8(c)). Note that many of the neighbors, despite not being vices, are some kind of device or instrument.

5.2 Results

We used a test set of 323 images, hand-picked so that the visual content was consistent with the text label. Using the voting tree described above, we classified them using $K = 80$ neighbors at a variety of semantic levels. To simplify the presentation of results, we collapsed the Wordnet tree by hand (which had 19 levels) down to 3 levels corresponding to one very high level ("organism", "object"), an intermediate level ("person", "plant", "animal") and a level typical of existing datasets ("fish", "bird", "herb").

In Fig. 9 we show the average ROC curve area for a classification task per word at each of the three semantic levels for D_1, D_2, D_3 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_3 outperforms the other distance metrics; (iii) the performance drops off as the classes become more specific.

In Fig. 10 we show the ROC curve area for a number of classes at different semantic levels, comparing the D_1 and D_3 metrics. For the majority of classes, D_3 may be seen to outperform D_1.

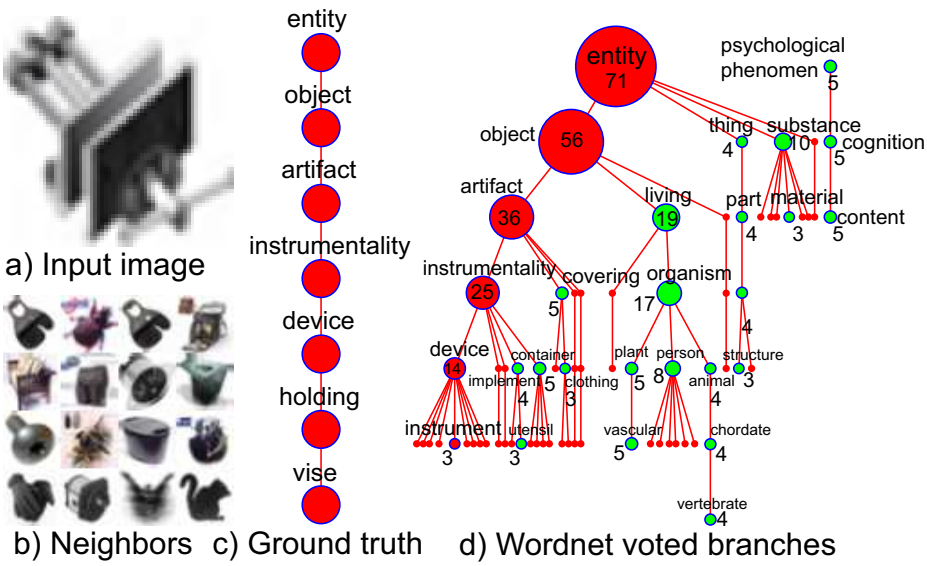

Figure 8. (a) Query image of a vise. (b) First 16 of 80 neighbors found using D_3. (c) Wordnet branch for vise. (d) Sub-tree formed by accumulating branches from all 80 neighbors. The red branch shows the nodes with the most votes. Note that this branch substantially agrees with the branch for vise.

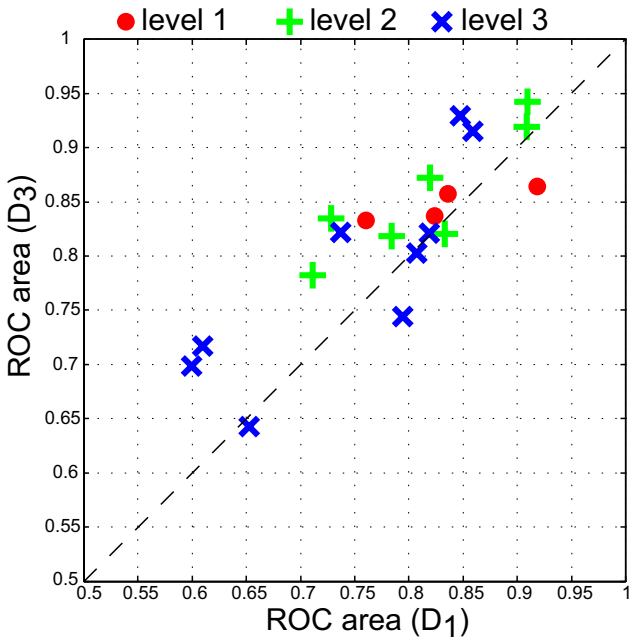

Figure 10. Scatter plot comparing D_1 and D_3 on classes from the semantic levels used in Fig. 9. In most cases D_3 beats D_1. Note the wide variation in performance of classes at the finer semantic level.

To illustrate the quality of the recognition achieved by using the $70,000,000$ weakly labeled images, we show in Fig. 11, 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 fine levels of categorization.

5.3 People

For certain classes the dataset is extremely rich. For example, many images on the web contain pictures of people. Thus for this class we are able to reliably find neighbors with similar locations, scales and poses to the query image, as shown in Fig. 12.

6 Conclusions

Many recognition tasks can be solved with images as small as 32×32 pixels. Working with tiny images has multiple benefits: features can be computed efficiently and collecting and working with larger collections of images becomes practical. We have presented a dataset with $70,000,000$ images, organized on a semantic hierarchy, and we have shown that, despite the dataset being weakly labeled, it can be effectively used for recognition tasks.

We have used simple nearest neighbor methods to obtain good recognition performance for a number of classes, such as "person". However, the performance of some other classes is poor (some of the classes in Fig. 10 have ROC curve areas around 65-70%). We believe that this is due to two shortcomings of our dataset: (i) sparsity of images in some classes; (ii) labelling noise. The former may be overcome by collecting more images, perhaps from sources other than the Internet. One approach to overcoming the labelling noise would be to bias the Wordnet voting toward images with high rank (using the performance curves obtained in Fig. 5(b)).

The dense sampling of categories provides an important dataset to develop transfer learning techniques useful for object recognition. Small images also present challenges

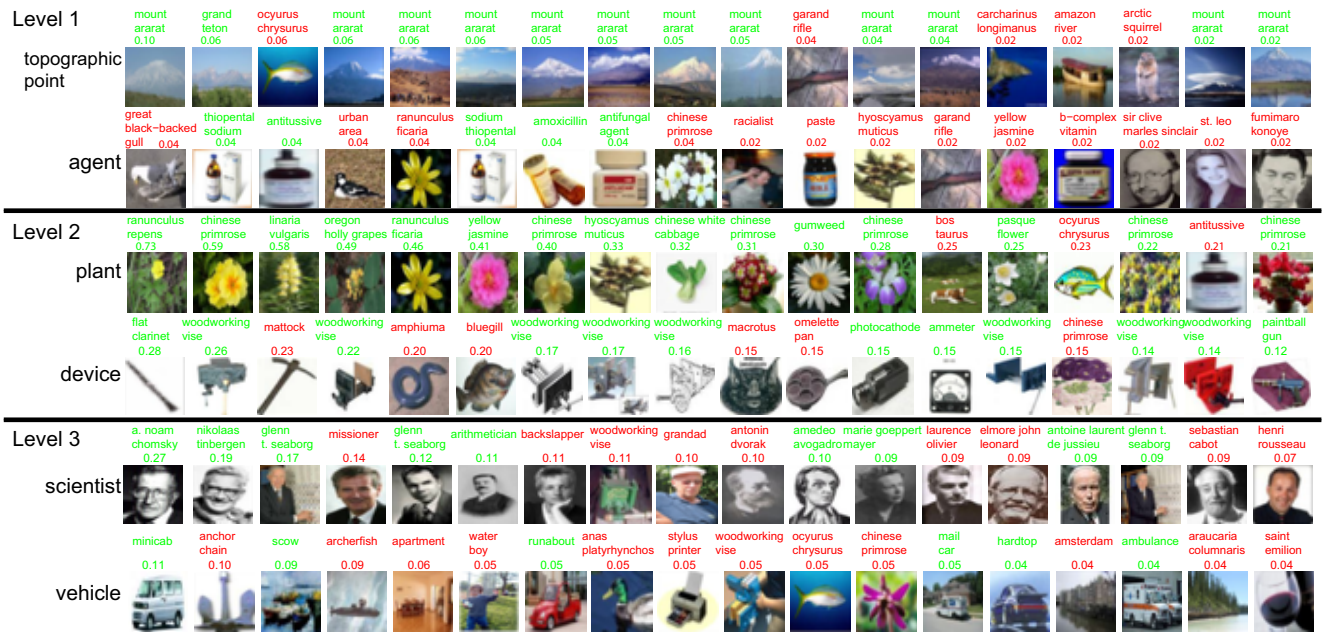

Figure 11. Test images assigned to words at each semantic level. The images are ordered by voting confidence from left to right (the confidence is shown above each image). The color of the text indicates if the image was correctly assigned (green) or not (red). The text on top of each image corresponds to the string that returned the image as a result of querying online image indexing tools. Each word is one of the 70,399 nouns from Wordnet.

for recognition - many objects can not be recognized in isolation. Therefore, recognition requires algorithms that incorporate contextual models, a direction for future work.

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

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, 3:1107–1135, Feb 2003. 6

[3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *Proc. CVPR*, volume 1, pages 26–33, June 2005. 2

[4] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y.-W. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. CVPR*, volume 2, pages 848–854, 2004. 4

[5] T. Berg and D. Forsyth. Animals on the web. In *Proc. CVPR*, pages 1463–1470, 2006. 4

[6] D. M. Chandler and D. J. Field. Estimates of the information content and dimensionality of natural scenes from proximity distributions. *JOSA*, 2006. 5

[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003. 2

[8] M. Everingham, A. Zisserman, C. K. I. Williams, and e. a. L. Van Gool. The 2005 pascal visual object classes challenge. In *LNAI 3944,*, pages 117–176, 2006. 3

[9] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, pages 524–531, 2005. 2

[10] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proc. ICCV*, volume 2, pages 1816–1823, Oct 2005. 4

[11] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Proc. ECCV*, pages 242–256, May 2004. 4

[12] K. Grauman and T. Darrell. Pyramid match hashing: Sublinear time indexing over partial correspondences. In *Proc. CVPR*, 2007. 2

[13] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report UCB/CSD-04-1366, California Institute of Technology, 2007. 4

[14] L. D. Harmon and B. Julesz. Masking in visual recognition: Effects of two-dimensional noise. *Science*, 180:1194–1197, 1973. 2

[15] A. Hoogs and R. Collins. Object boundary detection in images using a semantic ontology. In *AAAI*, 2006. 6

[16] 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. 2, 3

[17] D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, Sep 1999. 2

[18] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006. 5

[19] A. Oliva. Gist of the scene. In *Neurobiology of Attention, L. Itti, G. Rees & J. K. Tsotsos (Eds.)*, pages 251–256, 2005. 2

[20] A. Oliva and P. Schyns. Diagnostic colors mediate scene recognition. *Cognitive Psychology*, 41:176–210, 1976. 2

[21] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal in Computer Vision*, 42:145–175, 2001. 2, 3

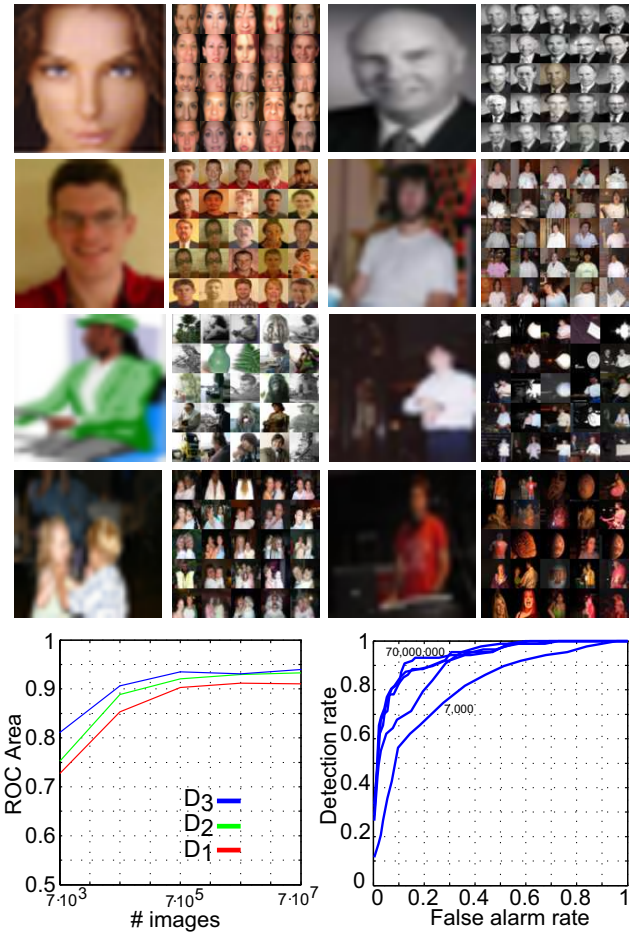

Figure 12. Some examples of test images belonging to the "person" node of the Wordnet tree, organized according to body size. Each pair shows the query image and the 25 closest neighbors out of $70,000,000$ images using D_3 with 32×32 images. Note how the neighbors match not just the category but also the location and size of the body in the image, which varies considerably in the examples. The bottom left figure shows the ROC curve area for the "person" node in the Wordnet tree as the number of images is varied, for D_1, D_2 and D_3. The bottom right figure shows the evolution of the ROC curve for "person" with D_3 as the number of images is varied.

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

[23] L. Renninger and J. Malik. When is scene recognition just texture recognition? *Vision Research*, 44:2301–2311, 2004. 2

[24] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. ICCV*, 2003. 2

[25] J. Wolfe. Visual memory: What do you know about what you saw? *Current Biology*, 8:R303–R304, 1998. 2

[26] Wordnet: a lexical database for the English language. Princeton Cognitive Science Laboratory, 2003. Version 2.0. 4, 6