Semantic Label Sharing for Learning with Many Categories

Rob Fergus[1], Hector Bernal[2], Yair Weiss[3], Antonio Torralba[2]

[1]Courant Institute, New York University
[2]CSAIL, MIT
[3]School of Computer Science Hebrew University

fergus@cs.nyu.edu, {hectorbernal,torralba}@csail.mit.edu, yweiss@cs.huji.ac.il

Abstract. In an object recognition scenario with tens of thousands of categories, even a small number of labels per category leads to a very large number of total labels required. We propose a simple method of *label sharing* between semantically similar categories. We leverage the WordNet hierarchy to define semantic distance between any two categories and use this semantic distance to share labels. Our approach can be used with any classifier. Experimental results on a range of datasets, upto 80 million images and 75,000 categories in size, show that despite the simplicity of the approach, it leads to significant improvements in performance.

1 Introduction

Large image collections on the Internet and elsewhere contain a multitude of scenes and objects. Recent work in computer vision has explored the problems of visual search and recognition in this challenging environment. However, all approaches require some amount of hand-labeled training data in order to build effective models. Working with large numbers of images creates two challenges: first, labeling a representative set of images and, second, developing efficient algorithms that scale to very large databases.

Labeling Internet imagery is challenging in two respects: first, the sheer number of images means that the labels will only ever cover a small fraction of images. Recent collaborative labeling efforts such as Peekaboom, LabelMe, ImageNet [2–4] have gathered millions of labels at the image and object level. However this is but a tiny fraction of the estimated 10 billion images on Facebook, let alone the hundreds of petabytes of video on YouTube. Second, the diversity of the data means that many thousands of classes will be needed to give an accurate description of the visual content. Current recognition datasets use 10's to 100's of classes which give a hopelessly coarse quantization of images into discrete categories. The richness of our visual world is reflected by the enormous number of nouns present in our language: English has around 70,000 that correspond to actual objects [5]. This figure loosely agrees with the 30,000 visual concepts estimated by psychologists [6]. Furthermore, having a huge number of classes dilutes the available labels, meaning that, on average, there will be relatively few

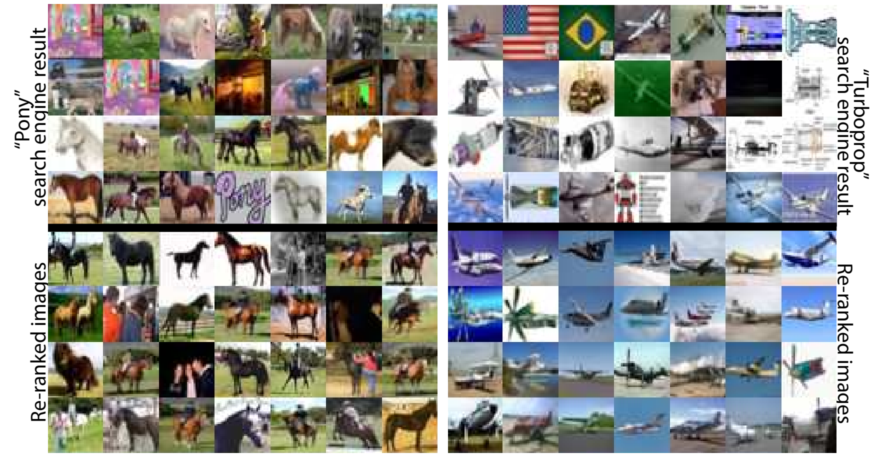

Fig. 1. Two examples of images from the Tiny Images database [1] being re-ranked by our approach, according to the probability of belonging to the categories "pony" and "turboprop" respectively. *No training labels were available for either class.* However 64,185 images from the total of 80 million were labeled, spread over 386 classes, some of which are semantically close to the two categories. Using these labels in our semantic label sharing scheme, we can dramatically improve search quality.

annotated examples per class (and many classes might not have any annotated data).

To illustrate the challenge of obtaining high quality labels in the scenario of many categories, consider the CIFAR-10 dataset constructed by Alex Krizhevsky and Geoff Hinton [7]. This dataset provides human labels for a subset of the Tiny Images [1] dataset which was obtained by querying Internet search engines with over 70,000 search terms. To construct the labels, Krizhevsky and Hinton chose 10 classes "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck", and for each class they used the WordNet hierarchy to construct a set of hyponyms. The labelers were asked to examine all the images which were found with a search term that is a hyponym of the class. As an example, some of the hyponyms of ship are "cargo ship", "ocean liner", and "frigate". The labelers were instructed to reject images which did not belong to their assigned class. Using this procedure, labels on a total of 386 categories (hyponyms of the 10 classes listed above) were collected at a cost of thousands of dollars.

Despite the high cost of obtaining these labels, the 386 categories are of course a tiny subset of the possible labels in the English language. Consider for example the words "pony" and "turboprop" (Fig. 1). Neither of these is considered a hyponym of the 10 classes mentioned above. Yet there is obvious information in the labeled data for "horse" and "airplane" that we would like to use to improve the search engine results of "pony" and "turboprop".

In this paper, we provide a very simple method for sharing labels between categories. Our approach is based on a basic assumption – we expect the classifier output for a single category to degrade gracefully with semantic distance. In other words, although horses are not exactly ponies, we expect a classifier for "pony" to give higher values for "horses" than to "airplanes". Our scheme, which we call "Semantic Label Sharing" gives the performance shown in Fig. 1. Even though we have no labels for "pony" and "turboprop" specifically, we can significantly improve the performance of search engines by using label sharing.

1.1 Related Work

Various recognition approaches have been applied to Internet data, with the aim of re-ranking, or refining the output of image search engines. These include: Li *et al.* [8], Fergus *et al.* [9], Berg *et al.* [10], amongst others. Our approach differs in two respects: (i) these approaches treat each class independently; (ii) they are not designed to scale to the billions of images on the web.

Sharing information across classes is a widely explored concept in vision and learning, and takes many different forms. Some of the first approaches applied to object recognition are based on neural networks in which sharing is achieved via the hidden layers which are common across all tasks [11, 12]. Error correcting output codes[13] also look at a way of combining multi-class classifiers to obtain better performance. Another set of approaches tries to transfer information from one class to another by regularizing the parameters of the classifiers across classes. Torralba *et al.* , Opelt *et al.* [14, 15] demonstrated its power in sharing useful features between classes within a boosting framework. Other approaches transfer information across object categories by sharing a common set of parts [16, 17], by sharing transformations across different instances [18–20], or by sharing a set of prototypes [21]. Common to all those approaches is that the experiments are always performed with relatively few classes. Furthermore, it is not clear how these techniques would scale to very large databases with thousands of classes.

Our sharing takes a different form to these approaches, in that we impose sharing on the class labels themselves, rather than in the features or parameters of the model. As such, our approach has the advantage that it it is independent of the choice of the classifier.

2 Semantic Label Sharing

Following [22] we define the semantic distance between two classes using a tree defined by WordNet[1]. We use a simple metric that measures the intersection between the ancestors of two words: the semantic distance S_{ij} between classes i and j (which are nodes in the tree) is defined as the number of nodes shared by their two parent branches, divided by the length of the longest of the two branches, i.e. $S_{ij} = \text{intersect}(par(i), par(j))/max(\text{length}(par(i)), \text{length}(par(j)))$, where

[1] Wordnet is graph-structured and we convert it into a tree by taking the most common sense of a word.

$par(i)$ is the path from the root node to node i. For instance, the semantic similarity between a "felis domesticus" and "tabby cat" is 0.93, while the distance between "felis domesticus" and a "tractor trailer" is 0.21. We construct a sparse semantic affinity matrix $A = \exp(-\kappa(1 - S))$, with $\kappa = 10$ for all the experiments in this paper. For the class "airbus", the nearest semantic classes are: "airliner" (0.49), "monoplane" (0.24), "dive bomber" (0.24), "twinjet" (0.24), "jumbo jet" (0.24), and "boat" (0.03). A visualization of A and a closeup are shown in Fig. 3(a) and (b).

Let us assume we have a total of C classes, hence A will be a $C \times C$ symmetric matrix. We are given L labeled examples in total, distributed over these C classes. The labels for class c are represented by a binary vector y_c of length L which has values 1 for positive hand-labeled examples and 0 otherwise. Hence positive examples for class c are regarded as negative labels for all other classes. $Y = \{y_1, \ldots, y_C\}$ is an $N \times C$ matrix holding the label vectors from all classes.

We share labels between classes by replacing Y with YA. This simple operation has a number of effects:

- Positive examples are copied between classes, weighted according to their semantic affinity. For example, the label vector for "felis domesticus" previously had zero values for the images of "tabby cat", but now these elements are replaced by the value 0.93.
- However, labels from unrelated classes will only deviate slightly from their original state of 0 (dependent on the value of κ).
- Negative labeled examples from classes outside the set of C are unaffected by A (since they are 0 across all rows of Y).
- Even if each class has only a few labeled examples, the multiplication by A will effectively pool examples across semantically similar classes, dramatically increasing the number that can be used for training, provided semantically similar classes are present amongst the set of C.

The effect of this operation is illustrated in two examples on toy data, shown in Fig. 2. These examples show good classifiers can be trained by sharing labels between classes, given knowledge of the inter-class affinities, even when no labels are given for the target class. In Fig. 2, there are 9 classes but label data is only given for 7 classes. In addition to the labels, the system also has access to the affinities among the 9 classes. This information is enough to build classification functions for the classes with no labels (Fig. 2(d) and (f)).

From another perspective, our sharing mechanism turns the original classification problem into a regression problem: the formerly binary labels in Y become real-values in YA. As such we can adapt many types of classifiers to minimize regression error rather than classification error.

3 Sharing in Semi-Supervised Learning

Semi-supervised learning is an attractive option in settings where very few training examples exist since the density of the data can be used to regularize the

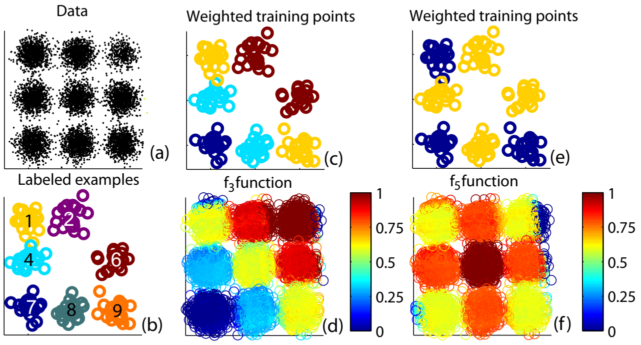

Fig. 2. Toy data illustrating our sharing mechanism between 9 different classes **(a)** in discrete clusters. For 7 of the 9 classes, a few examples are labeled **(b)**. No labels exist for the classes 3 and 5. **(c)**: Labels re-weighted by affinity to class 3. (Red=high affinity, Blue=low affinity). **(d)**: This plot shows the semi-supervised learning solution $f_{\text{class}=3}$ using weighted labels from **(c)**. The value of the function $f_{\text{class}=3}$ on each sample from **(a)** is color coded. Dark red corresponds to the samples more likely to belong to class 3. **(e)**: Labels re-weighted by affinity to class 5. **(d)**: Solution of semi-supervised learning solution $f_{\text{class}=5}$ using weighted labels from **(e)**.

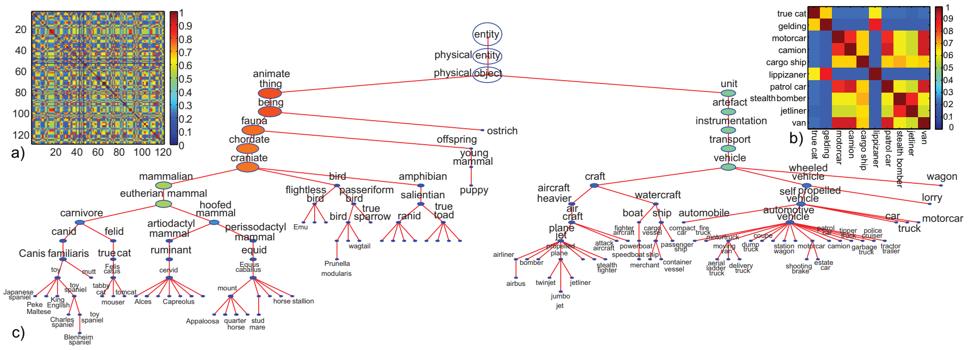

Fig. 3. Wordnet sub-tree for a subset of 386 classes used in our experiments. The associated semantic affinity matrix A is shown in **(a)**, along with a closeup of 10 randomly chosen rows and columns in **(b)**.

solution. This can help prevent over-fitting the few training examples and yield superior solutions. A popular class of semi-supervised algorithms are based on the graph Laplacian and we use an approach of this type.

We briefly describe semi-supervised learning in a graph setting. In addition to the L labeled examples $(X_l, Y_l) = \{(x_1, y_1), ..., (x_L, y_L)\}$ introduced above, we have an additional U unlabeled images $X_u = \{x_{L+1}, ..., x_N\}$, for a total of N images. We form a graph where the vertices are the images X and the edges are represented by an $N \times N$ matrix W. The edge weighting is given by $W_{ij} = \exp(-\|x_i - x_j\|^2/2\epsilon^2)$, the visual affinity between images i and j. Defining $D = diag(\sum_j W_{ij})$, we define the normalized graph Laplacian to be: $L = I = D^{-1/2}WD^{-1/2}$. We use L to measure the smoothness of solutions over the data points, desiring solutions that agree with the labels but are also smooth with respect to the graph. In the single class case we want to minimize:

$$J(f) = f^T L f + \sum_{i=1}^{l} \lambda (f_i - y_i)^2 = f^T L f + (f-y)^T \varLambda (f-y) \tag{1}$$

where $\varLambda$ is a diagonal matrix whose diagonal elements are $\varLambda_{ii} = \lambda$ if i is a labeled point and $\varLambda_{ii} = 0$ for unlabeled points. The solution is given by solving the $N \times N$ linear system $(L + \varLambda)f = \varLambda y$.

This system is impractical to solve for large N, thus it is common [23–25] to reduce the dimension of the problem by using the smallest k eigenvectors of L (which will be the smoothest) U as a basis with coefficients α: $f = U\alpha$. Substituting into Eqn. 1, we find the optimal coefficients α to be the solution of the following $k \times k$ system:

$$(\varSigma + U^T \varLambda U)\alpha = U^T \varLambda y \tag{2}$$

where $\varSigma$ is a diagonal matrix of the smallest k eigenvectors of L. While this system is easy to solve, the difficulty is computing the eigenvectors an $O(N^2)$ operation.

Fergus *et al.* [26] introduced an efficient scheme for computing approximate eigenvectors in $O(N)$ time. This approach proceeds by first computing numerical approximations to the eigenfunctions (the limit of the eigenvectors as $N \to \infty$). Then approximations to the eigenvectors are computed via a series of 1D interpolations into the numerical eigenfunctions. The resulting approximate eigenvectors (and associated eigenvalues) can be used in place of U and $\varSigma$ in Eqn. 2.

Extending the above formulations to the multi-class scenario is straightforward. In a multi-class problem, the labels will be held in an $N \times C$ binary matrix Y, replacing y in Eqn. 2. We then solve for the $N \times C$ matrix F using the approach of Fergus *et al.* Utilizing the semantic sharing from Section 2 is simple, with Y being replaced with YA.

4 Experiments

We evaluate our sharing framework on two tasks: (a) improving the performance of images returned by Internet search engines; (b) object classification. Note that the first problem consists of a set of 2-class problems (e.g. sort the pony images from the non-pony images), while the second problem is a multi-class classification with many classes.

These tasks are performed on three datasets linked to the Tiny Images database [1], a diverse and highly variable image collection downloaded from the Internet:

- *CIFAR:* This consists of 63,000 images from 126 classes selected[2] from the CIFAR-10 dataset [7], which is a hand-labeled sub-set of the Tiny Images.

[2] The selected classes were those that had at least 200 positive labels and 300 negative labels, to enable accurate evaluation.

These keywords and their semantic relationship to one another are shown in Fig. 3. For each keyword, we randomly choose a fixed test-set of 75 positive and 150 negative examples, reflecting the typical signal-to-noise ratio found in images from Internet search engines. From the remaining images for each class, we randomly draw a validation set of 25/50 +ve/-ve examples. The training examples consist of +ve/-ve pairs drawn from the remaining pool of 100 positive/negative images for each keyword.

- *Tiny:* The whole Tiny Images dataset, consisting of 79,302,017 images distributed over 74,569 classes (keywords used to download the images from the Internet). No human-provided labels are available for this dataset, thus instead we use the noisy labels from the image search engines. For each class we assume the first 5 images to be true positive examples. Thus over the dataset, we have a total of 372,845 (noisy) positive training examples, and the same number of negative examples (drawn at random). For evaluation, we can use labeled examples from either the *CIFAR* or *High-res* datasets.
- *High-res:* This is a sub-set of 10,957,654 images from the Tiny Images, for which the high-resolution original image exists. These images span 53,564 different classes, distributed evenly over all classes within the Tiny Images dataset. As with the *Tiny* dataset, we use no hand-labeled examples for training, instead using the first 5 examples for each class as positive examples (and 5 negative drawn randomly). For evaluation, we use 5,357 human-labeled images split into 2,569 and 2,788 positive and negative examples of each class respectively.

Pre-processing: For all datasets, each image is represented by a single Gist descriptor. In the case of the *Tiny* and *CIFAR* datasets, a 384-D descriptor is used which is then mapped down to 32 and 64 dimensions using PCA, for *Tiny* and *CIFAR* respectively. For the *High-res* dataset, a 512-D Gist descriptor is mapped down to 48-D using PCA.

4.1 Re-ranking experiments

On the re-ranking task we first use the *CIFAR* dataset to quantify the effects of semantic sharing. For each class separately we train a classifier on the training set (possibly using sharing) and use it to re-rank the 250 test images, measuring the precision at 15% recall. Unless otherwise stated, the classifier used is the semi-supervised approach of Fergus *et al.* [26].

In Fig. 4(left) we explore the effects of semantic sharing, averaging performance over all 126 classes. The validation set is used to automatically select the optimal values of κ and λ. The application of the Wordnet semantic affinity matrix can be seen to help performance. If the semantic matrix is randomly permuted (but with the diagonal fixed to be 1), then this is somewhat worse than not using sharing. But if the sharing is inverted (by replacing A with $1-A$ and setting the diagonal to 1), it clearly hinders performance. The same pattern of results can be see in Fig. 4(right) for a nearest neighbor classifier. Hence the

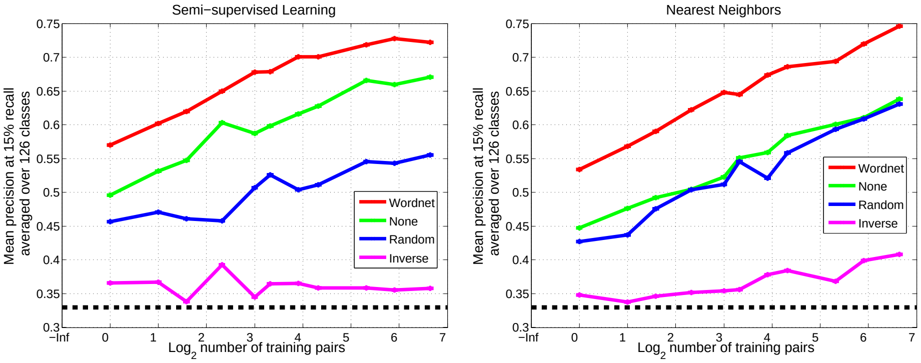

Fig. 4. *Left:* Performance for different sharing strategies with the semi-supervised learning approach of [26] as the number of training examples is increased, using 126 classes in the *CIFAR* dataset. *Right:* As for (left) but with a nearest neighbor classifier. The black dashed line indicates chance level performance. When the Wordnet matrix is used for sharing it gives a clear performance improvement (red) to both methods over no sharing [26] (green). However, if the semantic matrix does not reflect the similarity between classes, then it hinders performance (e.g. random (blue) and inverse (magenta) curves).

semantic matrix must reflect the relationship between classes if it is to be effective. In Fig. 5 we show examples of the re-ranking, using the semi-supervised learning scheme in conjunction with the Wordnet affinity matrix.

In Fig. 6(left & middle), we perform a more systematic exploration of the effects of Wordnet sharing. For these experiments we use fixed values of $\kappa = 5$ and $\lambda = 1000$. Both the number of classes and number of images are varied, and the performance recorded with and without the semantic affinity matrix. The sharing gives a significant performance boost, particularly when few training examples are available.

The sharing behavior can be used to effectively learn classes for which we have zero training examples. In Fig. 7, we explore what happens when we allocate 0 training images to one particular class (the left-out class) from the set of 126, while using 100 training pairs for the remaining 125 classes. When the sharing matrix is not used, the performance of the left-out class drops significantly, relative to its performance when training data is available (i.e. the point for each left-out class falls below the diagonal). But when sharing is used, the drop in performance is relatively small, with points being spread around the diagonal.

Motivated by Fig. 7, we show in Fig. 1 the approach applied to the *Tiny* dataset, using the human-provided labels from the *CIFAR* dataset. However, no CIFAR labels exist for the two classes selected (Pony, Turboprop). Instead, we used the Wordnet matrix to share labels from semantically similar classes for which labels do exist. The qualitatively good results demonstrated in Fig. 1 can only be obtained relatively close to the 126 keywords for which we have labels.

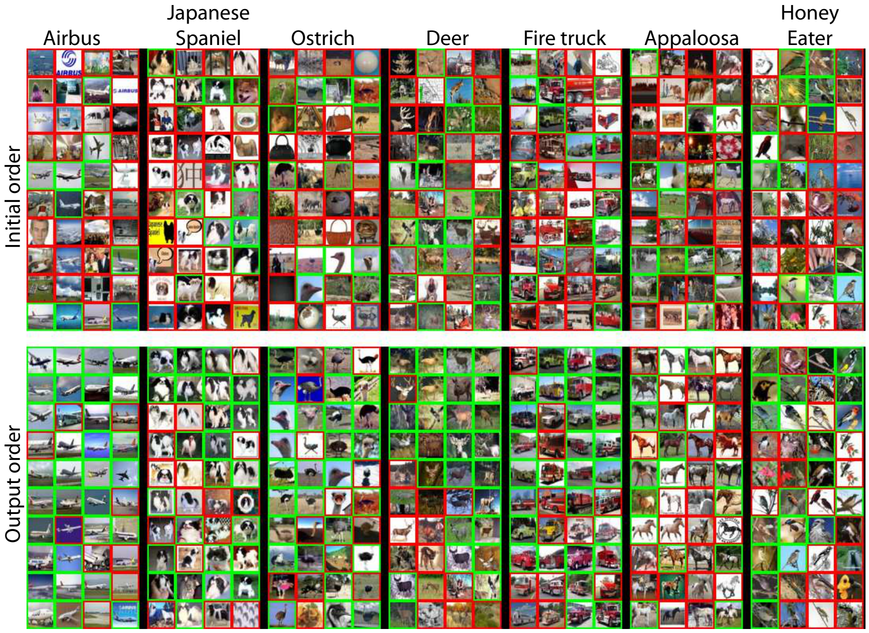

Fig. 5. Test images from 7 keywords drawn from the 126 class *CIFAR* dataset. The border of each image indicates its label (used for evaluation purposes only) with respect to the keyword, green = +ve, red = -ve. The top row shows the initial ranking of the data, while the bottom row shows the re-ranking of our approach trained on 126 classes with 100 training pairs/classes.

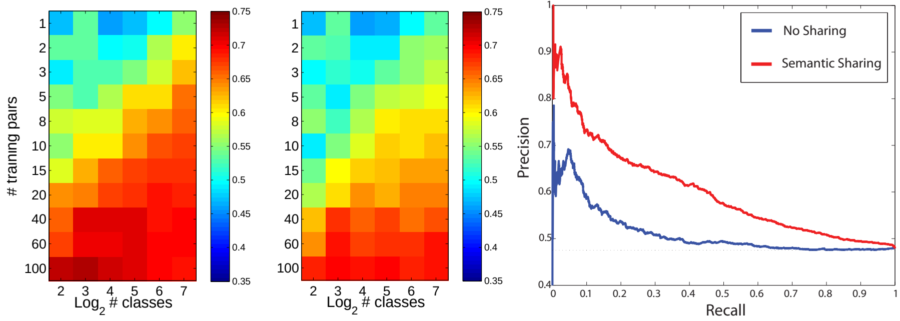

Fig. 6. *Left & Middle:* The variation in precision for the semi-supervised approach as the number of training examples is increased, using 126 classes with (left) and without (middle) Wordnet sharing. Note the improvement in performance for small numbers of training examples when the Wordnet sharing matrix is used. *Right:* Evaluation of our sharing scheme for the re-ranking task on the 10 million image *High-res* dataset, using 5,357 test examples. Our classifier was trained using 0 hand-labeled examples and 5 noisy labels per class. Using a Wordnet semantic affinity matrix over the 53,564 classes gives a clear boost to performance.

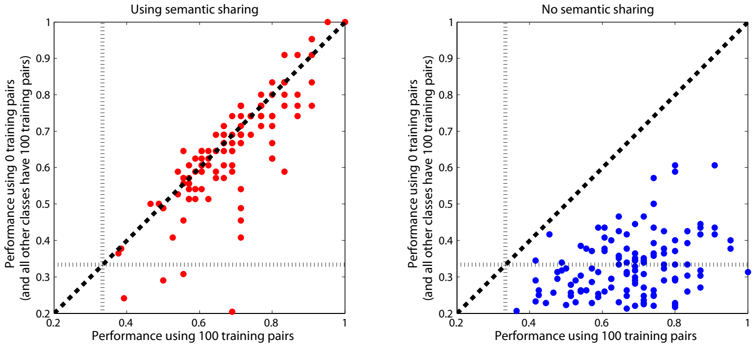

Fig. 7. An exploration of the performance with 0 training examples for a single class, if all the other classes have 100 training pairs. *Left:* By using the sharing matrix A, we can obtain a good performance by transferring labels from semantically similar classes. *Right:* Without it, the performance drops significantly.

This performance gain obtained by Wordnet sharing is quantified in a large-scale setting in Fig. 6(right) using the *High-res* dataset. Chance level performance corresponds to $2569/(2569+2788) = 48\%$. Without any sharing, the semi-supervised scheme (blue) gives a modest performance. But when the Wordnet sharing is added, there is significant performance boost.

Our final re-ranking experiment applies the semantic sharing scheme to the whole of the *Tiny* dataset (with no CIFAR labels used). With 74,569 classes, many will be very similar visually and our sharing scheme can be expected to greatly assist performance. In Fig. 11 we show qualitative results for 4 classes. The semi-supervised algorithm takes around 0.1 seconds to perform each re-ranking (since the eigenfunctions are precomputed), compared to over 1 minute for the nearest-neighbor classifier. These figures show qualitatively that the semi-supervised learning scheme with semantic sharing clearly improves search performance over the original ranking and that without the sharing matrix the performance drops significantly.

4.2 Classification experiments

Classification with many classes is extremely challenging. For example, picking the correct class out of 75,000 is something that even humans typically cannot do. Hence instead of using standard metrics, we measure how far the predicted class is from the true class, as given by the semantic distance matrix S. Under this measure the true class has distance 0, while 1 indicates total dissimilarity. Fig. 8 illustrates this metric with two example images and a set of samples varying in distance from them.

We compare our semantic sharing approach in the semi-supervised learning framework of [26] to two other approaches: (i) linear 1-vs-all SVM; (ii) the hierarchical SVM approach of Marszalek and Schmid [27]. The latter method uses the semantic relationships between classes to construct a hierarchy of SVMs. In

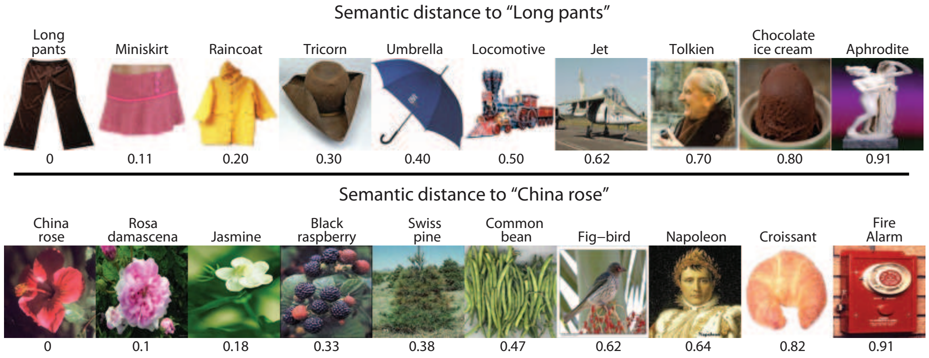

Fig. 8. Our semantic distance performance metric for two examples "Long pants" and "China rose". The other images are labeled with their semantic distance to the two examples. Distances under 0.2 correspond to visual similar objects.

implementing this approach, we use the same Wordnet tree structure from which the semantic distance matrix S is derived. At each edge in the tree, we train a linear SVM in the manner described in [27]. Note that both our semantic sharing method and that of Marszalek and Schmid are provided with the same semantic information. Hence, by comparing the two approaches we can see which makes more efficient use of the semantic information.

These three approaches are evaluated on the *CIFAR* and *High-res* datasets in Figures 9 and 10 respectively. The latter dataset also shows the semi-supervised scheme without sharing. The two figures show consistent results that clearly demonstrate: (i) the addition of semantic information helps – both the H-SVM and SSL with sharing beat the methods without it; (ii) our sharing framework is superior to that of Marszalek and Schmid [27].

5 Summary and future work

We have introduced a very simple mechanism for sharing training labels between classes. Our experiments on a variety of datasets demonstrate that it gives significant benefits in situations where there are many classes, a common occurrence in large image collections. We have shown how semantic sharing can be combined with simple classifiers to operate on large datasets up to 75,000 classes and 79 million images. Furthermore, our experiments clearly demonstrate that our sharing approach outperforms other methods that use semantic information when constructing the classifier. While the semantic sharing matrix from Wordnet has proven effective, a goal of future work would be to learn it directly from the data.

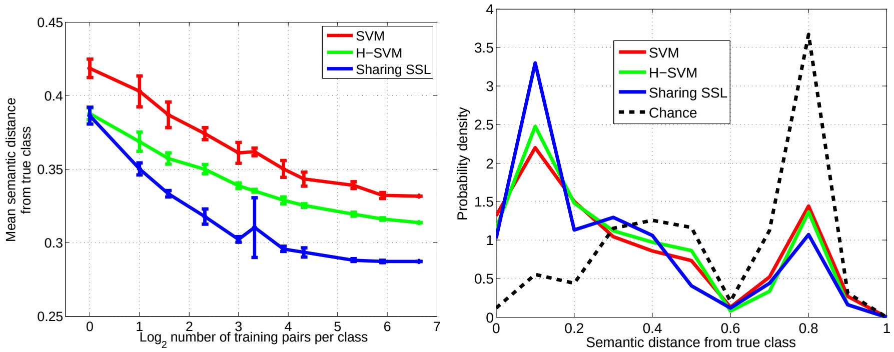

Fig. 9. Comparison of approaches for classification on the *CIFAR* dataset. Red: 1 vs all linear SVM; Green: Hierarchical SVM approach of Marszalek and Schmid [27]; Blue: Our semantic sharing scheme in the semi-supervised approach of [26]; Black: Chance. *Left:* Mean semantic distance of test examples to true class as the number of labeled training examples increases (smaller is better). *Right:* For 100 training examples per class, the distribution of distances for the positive test examples. Our sharing approach has a significantly lower mean semantic distance, with a large mass at a distance < 0.2, corresponding to superior classification performance. See Fig. 8 for an illustration of semantic distance.

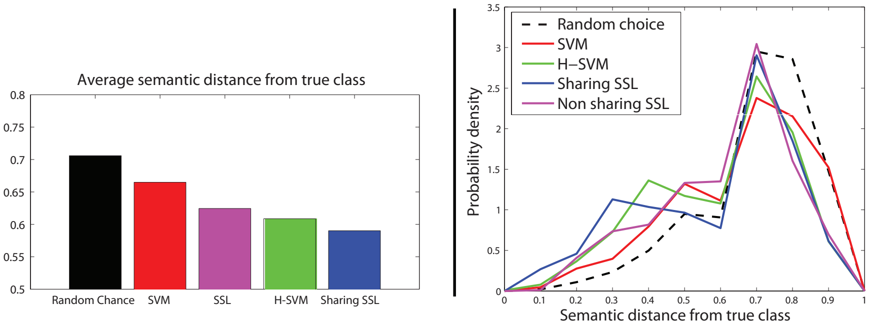

Fig. 10. Comparison of approaches for classification on the *High-res* dataset. Red: 1 vs all linear SVM; Green: Hierarchical SVM approach of Marszalek and Schmid [27]; Magenta: the semi-supervised scheme of [26]; Blue: [26] with our semantic sharing scheme; Black: Random chance. *Left:* Bar chart showing mean semantic distance from true label on test set. *Right:* The distribution of distances for each method on the test set. Our approach has more mass at a distance < 0.2, indicating superior performance.

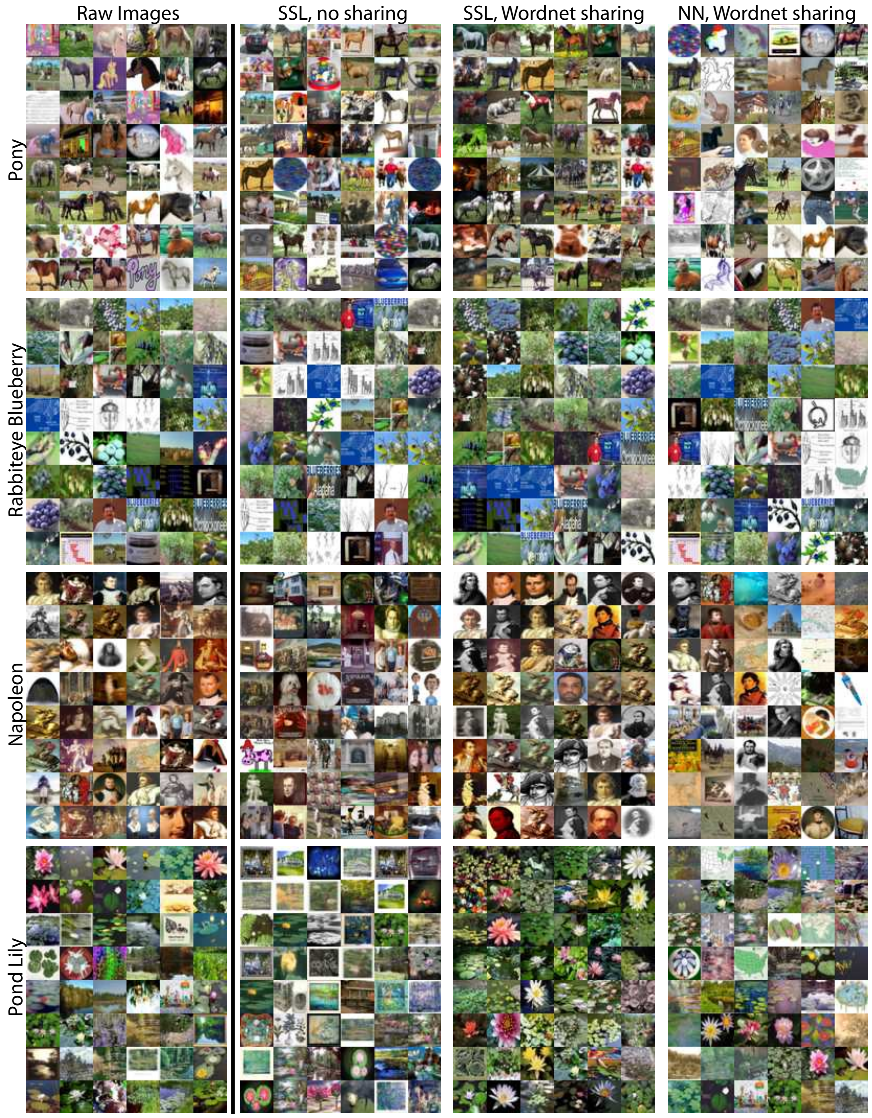

Fig. 11. Sample results of our semantic label sharing scheme on the *Tiny* dataset (79 million images). 0 hand-labeled training examples were used. Instead, the first 5 images of each of the 74,569 classes were taken as positive examples. Using these labels, classifiers were trained for 4 different query classes: "pony", "rabbiteye blueberry", "Napoleon" and "pond lily". Column 1: the raw image ranking from the Internet search engine. Column 2: re-ranking using the semi-supervised scheme without semantic sharing. Column 3: re-ranking with semi-supervised scheme and semantic sharing. Column 4: re-ranking with a nearest-neighbor classifier and semantic sharing. Without semantic sharing, the classifier only has 5 positive training examples, thus performs poorly. But with semantic sharing it can leverage the semantically close examples from the pool of 5*74,569=372,845 positive examples.

References

1. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large database for non-parametric object and scene recognition. IEEE PAMI **30** (2008) 1958–1970
2. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. IJCV **77** (2008) 157–173
3. van Ahn, L.: The ESP game (2006)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09. (2009)
5. Fellbaum, C.: Wordnet: An Electronic Lexical Database. Bradford Books (1998)
6. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review **94** (1987) 115–147
7. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
8. Li, L.J., Wang, G., Fei-Fei, L.: Imagenet. In: CVPR. (2007)
9. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV. Volume 2. (2005) 1816–1823
10. Berg, T., Forsyth, D.: Animals on the web. In: CVPR. (2006) 1463–1470
11. Caruana, R.: Multitask learning. Machine Learning **28** (1997) 41–75
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86** (1998) 2278–2324
13. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via ECOCs. JAIR **2** (1995) 263–286
14. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proc. of the 2004 IEEE CVPR. (2004)
15. Opelt, A., Pinz, A., Zisserman, A.: Incremental learning of object detectors using a visual shape alphabet. In: CVPR (1). (2006) 3–10
16. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence (To appear - 2004)
17. Sudderth, E., Torralba, A., Freeman, W., Willsky, A.: Learning hierarchical models of scenes, objects, and parts. In: Proceedings of the IEEE International Conference on Computer Vision, Beijing. (2005) To appear
18. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural Computation **12** (2000) 1247–1283
19. Bart, E., Ullman, S.: Cross-generalization: learning novel classes from a single example by feature replacement. In: CVPR. (2005)
20. Miller, E., Matsakis, N., Viola, P.: Learning from one example through shared densities on transforms. In: CVPR. Volume 1. (2000) 464–471
21. Quattoni, A., Collins, M., Darrell, T.: Transfer learning for image classification with sparse prototype representations. In: CVPR. (2008)
22. Budanitsky, Hirst: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics (2006)
23. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press (2006)
24. Schoelkopf, B., Smola, A.: Learning with Kernels Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, (2002)
25. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: In ICML. (2003) 912–919
26. Fergus, R., Weiss, Y., Torralba, A.: Semi-supervised learning in gigantic image collections. In: NIPS. (2009)
27. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. In: CVPR. (2007)