Measuring Image Distances via Embedding in a Semantic Manifold

Chen Fang and Lorenzo Torresani

Dartmouth College, Computer Science Department
Hanover, NH, USA
http://vlg.cs.dartmouth.edu

Abstract. In this work we introduce novel image metrics that can be used with distance-based classifiers or directly to decide whether two input images belong to the same class. While most prior image distances rely purely on comparisons of low-level features extracted from the inputs, our metrics use a large database of labeled photos as auxiliary data to draw semantic relationships between the two images, beyond those computable from simple visual features. In a preprocessing stage our approach derives a semantic image graph from the labeled dataset, where the nodes are the labeled images and the edges connect pictures with related labels. The graph can be viewed as modeling a semantic image manifold, and it enables the use of graph distances to approximate semantic distances. Thus, we reformulate the task of measuring the semantic distance between two *unlabeled* pictures as the problem of embedding the two input images in the semantic graph. We propose and evaluate several embedding schemes and graph distance metrics. Our results on Caltech101, Caltech256 and ImageNet show that our distances consistently match or outperform the state-of-the-art in this field.

1 Introduction

Psychological studies have shown that humans can easily determine whether two visual examples belong to the same basic category, even when that class is new and has never been seen before [1]. This suggests that to address this problem our brain employs a general semantic distance metric valid across all classes. In this work we are interested in investigating computational models that can tackle the same problem: our objective is to design distance functions providing a measure of whether two input photos belong to the same basic class, regardless of what that class may be. Image metrics implementing such semantic notions of similarity promise to enable a wide array of computer vision applications, and have been used in the past in image retrieval [2, 3], object classification [4], as well as semantic segmentation and annotation of photos [5].

Most prior image metrics rely solely on comparisons of low-level features extracted from the two input images [2, 4, 6]. While directly comparing visual features may be sufficient to assess simple notions of similarity, such as near-duplicate or object-instance similarity, we argue for the need of auxiliary labeled

data to provide accurate estimates of semantic relatedness and membership to the same object class. In a sense, our proposed use of this background knowledge is akin to the way children exploit past observations of many examples of different classes in order to learn to recognize instances of a new category [7].

At a high-level, our approach operates as follows. During an offline preprocessing stage, our method uses the auxiliary dataset of images with class labels to compute an image graph. The nodes of the graph are the labeled pictures and the edges link images that are semantically similar, as determined by the class labels. The shortest path distance between two nodes of the graph can then be viewed as a measure of their semantic relatedness. The shortest path distances are approximations of the geodesic distances of the unknown semantic manifold of images. This idea has been previously used for many tasks including dimensionality reduction [8], and semi-supervised learning [9]. While the graph per se provides estimates of semantic distance only for the labeled image nodes, we propose to extend its use also for unlabeled pictures, by embedding these photos in the graph. For each unlabeled input photo, this requires first determining its position in the graph, using only visual features. Once the input is embedded in the graph, we can compute its semantic distance to all the other pictures in the graph. Similarly, given two unlabeled images, we can embed them both in the graph in order to measure their semantic distance.

In this paper we propose several schemes to embed unlabeled pictures in the semantic graph. Our methods perform the embedding by using a visual distance (i.e., a metric based on low-level image features) to compare the input photo to the images in the graph. While this may appear to defeat the purpose of sidestepping visual distances to measure semantic relationships between images, we argue that our embedding task is far easier than the problem of directly computing semantic distances from low-level descriptors, for the following reasons:

1. Embedding the input images requires only selecting the most semantically-related photos. As shown by studies in human perception [10] and in computer vision [11], the most semantically similar pictures to a given input photo tend to be those most visually similar to it. Thus, these images are easy to identify even with distances based on low-level image descriptors.
2. We can simplify the task by using a large-scale database of labeled photos. It has been shown [12] that making the database larger will increase the probability that the top images retrieved according to a visual metric will also be semantically close to the input image, even when using simple low-level features to calculate visual distances. We exploit this property by using a database of 10M images (the ImageNet dataset [13]) to build our graph.
3. It is possible to exploit the structure of the graph to improve the embedding results: while the visual distances are brittle and may produce a set of candidate nodes including some outliers (i.e., images not semantically related to the input photo), this candidate set can then be refined (or denoised) to identify related nodes that lie close to each other in the graph. In other words, it is possible to enforce coherence of labels among the selected nodes to make the embedding more accurate and robust.

2 Related Work

Most object categorization systems require some form of similarity function to compare examples, such as the distance metric used by the nearest neighbor (NN) classifier or the kernels in SVMs. Most recent approaches to defining image metrics are based on learning methods which train the distance function using a set of labeled examples, typically consisting of images annotated with class labels. This problem is often referred to as metric learning. Within the wide range of proposed approaches in this area we can identify two main categories: techniques to learn "global" metrics versus methods computing "local" distances.

Algorithms in the former category operate by learning a single parametric transformation mapping the inputs to a new target space, such that a predefined metric (most typically the Euclidean distance) in this space satisfies certain desired properties [14, 15]. Similarly to these approaches, our method uses labeled examples to map images to a new target space – in our case, the semantic graph. However, rather than computing a parametric transformation and employing a predefined distance in the target space, our method uses the examples non-parametrically both to compute the mapping and also to define a distance metric expressed in terms of the entire labeled set.

The second strand of related work involves methods to compute "local" distances, i.e., metrics that vary across the space of examples (see [6] for a comprehensive survey). A simple form of local distance is one that changes for each individual training example [2, 4]. Alternatively, a local distance can be learned for each category to recognize [16] or by grouping together classes that can share effectively the same metric [17]. Our approach can be viewed also as implementing a local metric, since the semantic graph can be complex and anisotropic: our distance will vary depending on the embedding point of the test example. In a sense, our metric is closely related to algorithms that learn a different metric for each *test* example by using as training points its closest neighbors [18, 19]. However, unlike these prior systems, we exploit label information associated to the training examples, so as to suppress the effect of outliers present in the visual neighbors and to obtain a distance that is optimized for class recognition. Furthermore, while prior local metrics have been trained for a predefined set of classes (with the only exception of [17] which demonstrates good generalization to novel classes), our aim is to define a general distance that can be used to compare images of arbitrary classes, even categories not present in the labeled graph. Indeed, nearly all our experiments are carried out with this setup.

Our approach is inspired by the recent work of Deselaers and Ferrari [11], who have also proposed to compute image distances through comparisons to an auxiliary labeled dataset. They named their metric the "ImageNet distance", as it relies on the ImageNet database to infer the semantic relation between the input photos. For each input image, their method computes the distribution of class labels associated to its ImageNet neighbors; the distance between two input images is then calculated by comparing their class-label histograms. A related idea is presented in [3] where the distance between two images is computed by

comparing their membership probabilities to a set of 103 Flickr groups, estimated using a set of SVM classifiers.

As in [11], we also exploit ImageNet as the auxiliary source of labeled images. However, we argue that our graph-based representation of this data provides several advantages over the system of [11]. First, it enables semantic filtering: while the ImageNet distance uses the labels of *visual* neighbors to measure similarity between two images, our embedding methods exploit the graph structure to find target nodes that are not only visually similar to the input but also *semantically* coherent. In our experiments we demonstrate that this refinement improves the results. Furthermore, the graph allows us to measure indirect semantic relations: while the ImageNet distance measures the number of *exactly matching* class labels between the two neighbor sets, the graph allows us to take into account indirect semantic relations between the neighbors, even when their class labels do not match exactly.

3 Approach Overview

Our approach consists of an offline preprocessing stage, during which the semantic graph is built from a dataset of labeled images, and a test stage in which the graph is used to measure the distance between any two new unlabeled images.

Let us denote with $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ the labeled dataset of N images that we use to build the semantic graph, where $\mathbf{x}_i$ is the descriptor of the i-th image in the database and y_i is a label indicating the category of the object present in the i-th image. While our approach can be used with any arbitrary image descriptor, our experiments use two different feature vectors: the first is the GIST descriptor [20], which is a low-level image representation capturing the spatial layout in the picture; the second is the "classeme" feature vector [21], which is a higher-level descriptor containing the output of 2659 predefined classifiers evaluated on the image. We chose these two descriptors for several reasons: they are compact in size and thus well suited to large-scale databases; both descriptors have been shown to capture categorical information; finally, they allow us to understand the pros and cons of using a low-level as opposed to a high-level representation with our approach.

The final goal of our system is to compute the distance between any two unlabeled images $\mathbf{x}, \mathbf{x}'$. While sections 4,5 describe formally the method, here we explain the intuition behind the two stages of our approach, schematically illustrated in figure 1.

Offline stage: construction of the semantic graph. The aim of this stage is to reorganize the auxiliary dataset $\mathcal{D}$ in the form of a graph. The nodes in the graph represent the labeled images and the edges link pictures that are detected to be highly similar, *both visually as well semantically*. The semantic relation between the labeled images is determined by comparing their annotations, while their visual similarity is computed using image-content features. The edges in the graph are supplemented with weights, corresponding to the visual distance between the two nodes connected by the edge. The high-level idea is that for

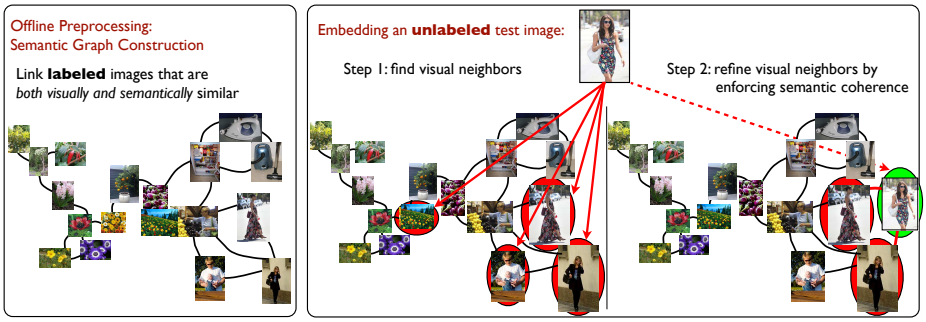

Fig. 1. Conceptual illustration of our embedding method. During an offline stage a semantic image graph is constructed using a labeled database: links are created between images that satisfy the joint conditions of being visually close and having related class labels. At test time, the unlabeled photo is embedded in the graph via a two-step process: first, visual neighbors are found; then, the position of the test images in the graph is computed by finding visual neighbors that are semantically coherent.

images that are closely related, the visual distance provides a reliable estimate of their similarity. For two images that are not directly linked via an edge, their similarity can be measured through their shortest connecting path within the graph. Thus, the graph embodies a form of semantic manifold where geodesics provide measures of semantic relatedness between images.

Test time: embedding unlabeled images in the graph. The semantic distance between two unlabeled images is computed by embedding independently both images in the graph so as to measure their distance in the semantic manifold. As illustrated in figure 1, this is done via the following two steps:

- *Step 1: find visual neighbors in the graph.* For each of the two input examples, the m closest database images are found according to the visual distance.
- *Step 2: embed the points by enforcing semantic coherence.* While the initial selection of the m candidate nodes ensures that these images are visually similar to the input, in this step we impose semantic coherence among these nodes to compute the final positions of the inputs in the graph.

After embedding, the distance between the inputs is calculated by comparing their positions inside the semantic graph.

4 Semantic Graph Construction

As previously discussed, we construct our semantic graph from the large-scale ImageNet dataset [13], which consists of roughly 10M images encompassing over 15000 categories. The ImageNet categories are structured according to the semantic hierarchy of WordNet [22]: each class is described by a set of synonyms, called a *synset*, and the children of a synset represent more specialized synsets of

that visual category (e.g., the children of synset "plant" are "tree", "flower" and "vegetable"). For each synset, on average, the dataset includes 632 manually-validated images illustrating that visual concept.

We exploit the hierarchy of ImageNet to build the semantic graph, as discussed below. In order to maintain the computational and storage costs manageable in spite of the large database size, we propose to construct a sparse graph, where each image is connected only to a small number of other photos.

For each database image $\mathbf{x}_i$, we define $\mathcal{S}_i$ to be the set of synsets comprising the synset of $\mathbf{x}_i$ and the children of the synsets of $\mathbf{x}_i$. We refer to $\mathcal{S}_i$ as the "extended synset" of $\mathbf{x}_i$. Then, the graph is constructed by creating an undirected edge between each image $\mathbf{x}_i$ and its k-closest neighbors within its extended synset $\mathcal{S}_i$, computed using the L2 distance between image descriptors. To each edge connecting node i to node j, we associate weight $w_{ij} \equiv ||\mathbf{x}_i - \mathbf{x}_j||$. Note that this strategy achieves two fundamental goals: on one hand, by linking each image only to nodes within its extended synset we establish semantically-consistent edges; on the other hand, by letting edges to be created across the original WordNet synsets, we avoid ending up with a myriad of disconnected graph components.

One issue, however, is that the root node in the ImageNet hierarchy has no associated images. This would cause the subtrees of the top-layer synsets to be disconnected components in the graph. To avoid this problem, we establish edges between the image pairs with the 1000 smallest visual distances among all pictures in the top synsets. After this operation, 99.36% of all images belong to the largest connected component of the graph. Thus, we simply discard the images outside the largest component, since this is a tiny subset of the database.

5 Embedding Unlabeled Images in the Graph

We now present different strategies to embed an unlabeled image $\mathbf{x}$ in the graph. The initial step for all methods involves selecting a set of candidate neighbors in the graph using the visual distance: we indicate with $\mathcal{R} \subset \{1, \ldots, N\}$ the indices of the m-nearest neighbors of $\mathbf{x}$ in the graph, computed according to the L2 distance between image descriptors. In practice, the set $\mathcal{R}$ will include images semantically related to $\mathbf{x}$ but also some outliers. The methods described below enforce semantic coherence to improve the embedding.

Semantic Energy Optimization (SEO). This embedding method operates by connecting the input to a subset of n nodes $\mathcal{T}$, which we name the *target nodes*. The subset $\mathcal{T}$ is chosen from the set of visual neighbors $\mathcal{R}$ by imposing semantic coherence via an energy optimization approach. While the parameter n could be set a-priori to be equal to k, in practice we found beneficial to tune n via cross validation. We represent the subset $\mathcal{T} \subset \mathcal{R}$ by introducing binary variables $z_i \in \{0, 1\}$ for the nodes $i \in \mathcal{R}$: we use $z_i = 1$ to indicate that $i \in \mathcal{T}$ (i.e., the node is selected as a target node), while $z_i = 0$ denotes that $i \notin \mathcal{T}$. We indicate with $\mathbf{z}$ the $|\mathcal{R}|$-dimensional binary-valued vector obtained by concatenating these binary variables, i.e., $\mathbf{z} = (z_i \mid i \in \mathcal{R})$. An intuitive idea is to determine the subset

$\mathcal{T}$ by minimizing the following energy function:

$$E(\mathbf{z}) = \sum_{i \in \mathcal{R}} \theta_i z_i + \lambda \sum_{i,j \in \mathcal{R}} \theta_{ij} z_i z_j \tag{1}$$

subject to constraint $\sum_{i \in \mathcal{R}} z_i = n$, where

$$\theta_i = ||\mathbf{x} - \mathbf{x}_i|| \tag{2}$$

$$\theta_{ij} = \begin{cases} d^{SPD}(\mathbf{x}_i, \mathbf{x}_j) & \text{if } d^{SPD}(\mathbf{x}_i, \mathbf{x}_j) < \tau \\ \tau & \text{otherwise} \end{cases} \tag{3}$$

with $d^{SPD}(\mathbf{x}_i, \mathbf{x}_j)$ denoting the shortest path distance in the semantic graph between $\mathbf{x}_i$ and $\mathbf{x}_j$. Intuitively, the unary terms of eq. 1 encode our preference for choosing nodes that are visually similar to $\mathbf{x}$, while the pairwise terms encourage selection of neighbors that are close to each other in the semantic graph. The threshold τ is used to avoid penalizing excessively selection of nodes that are far apart in the graph: this makes the model more robust to outliers. It can be shown that the constrained discrete optimization defined by eq. 1 is NP-hard in general [23]. Nevertheless, we have tried to minimize this energy by reformulating the optimization as a mixed integer program (MIP) expressed in term of auxiliary variables $t_{ij} \in [0, 1]$ bounding the pairwise interactions $z_i z_j$ via constraints $t_{ij} \leq z_i$, $t_{ij} \leq z_j$, $t_{ij} \geq z_i + z_j - 1$ for all $i, j \in \mathcal{R}$. We obtained good optimization results by minimizing the resulting MIP with the state-of-the-art Gurobi solver [24], which in practice globally optimizes 26% of our problems.

However, even when the optimal $\mathcal{T}$ could be found, the resulting embedding did not perform well in our tests. Through experimental investigation, we discovered that the energy model of eq. 1 is simply too strict as it wants *all* target neighbors to be close to each other. In practice, for many images this is an unreasonable assumption. Consider for example the photo of a group of children playing soccer in the street: the picture should be linked to nodes of synset "soccer, association football" but possibly also to nodes of synset "city, metropolis, urban center". Based on this observation we designed a "softer" version of our semantic energy that forces each selected node to be close to at least l other target nodes, where $l < n$. In other words, we encourage each selected neighbor to be near a few other target nodes, but not necessarily to *all* nodes in $\mathcal{T}$. This soft constraint is implemented by optimizing the following energy:

$$E(\mathbf{z}, \mathbf{t}) = \sum_{i \in \mathcal{R}} \theta_i z_i + \lambda \sum_{i,j \in \mathcal{R}} \theta_{ij} t_{ij} \tag{4}$$

subject to $\sum_{i \in \mathcal{R}} z_i = n$, and to constraints:

$$z_i \in \{0, 1\},\ t_{ij} \in \{0, 1\},\ t_{ij} \leq z_i, t_{ij} \leq z_j\ \forall i, j \in \mathcal{R} \tag{5}$$

$$\sum_{j \in \mathcal{R}} t_{ij} \geq l z_i\ \forall i \in \mathcal{R} \tag{6}$$

where θ_i, θ_{ij} are defined as above. These constraints ensure that for each target node i, only the l smallest pairwise terms θ_{ij} between i and other selected nodes

are included in the objective. We found that this optimization is also much easier to solve: the Gurobi solver was able to globally optimize *all* of our test cases. We refer to minimization of eq. 4 as Semantic Energy Optimization (SEO).

Random Walk (RW). The high-level idea of this embedding method is to find the nodes that are most likely to be reached by a Markov random walk [25] inside the graph starting from the initial candidate nodes $\mathcal{R}$. We use random walks to denoise the initial set $\mathcal{R}$ by finding nodes that are "close" to the majority of these initial vertices, while suppressing the effect of the outliers in $\mathcal{R}$. In order to perform the random walk, for each graph edge linking i to j we define the one-step transition probability from i to j in terms of the weights w_{ij} (we remind the reader that the weights w_{ij} are the visual distances computed during the graph construction). Specifically, for each edge (i, j) we define the probability of transitioning from node i at time t to node j at time $t + 1$ to be

$$P_{t+1|t}(j|i) = \frac{1/w_{ij}}{\sum_k 1/w_{ik}} \tag{7}$$

so that the probabilities out of node i sum up to 1 (this probability is set to 0 for nodes not directly connected by an edge). Note that for nodes linked by an edge $P_{t+1|t}(j|i)$ is inversely proportional to the visual distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. This implies that at each time the walk is likely to progress into a node that is highly similar to the current one. The random walk is initiated from a starting distribution $\mathbf{q} \in \mathbb{R}^N$ computed from the candidate nodes $\mathcal{R}$ as follows:

$$q_i = \begin{cases} \frac{1/||\mathbf{x}-\mathbf{x}_i||}{\sum_{k\in\mathcal{R}} 1/||\mathbf{x}-\mathbf{x}_k||} & \text{if } i \in \mathcal{R} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

If we store the one-step probabilities into a matrix A whose (i, j)-th entry is set equal to $P_{t+1|t}(j|i)$, then we can calculate the distribution $\mathbf{r}$ of nodes reached from $\mathbf{q}$ after t steps of random walk as $\mathbf{r}^T = \mathbf{q}^T A^t$. This can be more efficiently calculated by means of t matrix-vector products, i.e., $\mathbf{r} = (((\mathbf{q}^T A)A) \dots A)$. The resulting vector $\mathbf{r}$ can be shown [26] to measure the "volume" of paths leading to the individual nodes in the graph from the initial configuration $\mathbf{q}$. Intuitively, the random walk will tend to suppress paths originating from outlier nodes in $\mathcal{R}$, while paths starting from nearby nodes in $\mathcal{R}$ will tend to reinforce each other.

In principle, we could select as target nodes $\mathcal{T}$ the vertices that correspond to the n largest entries in $\mathbf{r}$, i.e., the ones that are more likely to be reached from the initial configuration. However, we found this strategy to produce relatively poor results. Instead we have had more success by directly using the random walk probabilities r_i to calculate semantic distances as follows. Let $\mathbf{r}$ and $\mathbf{r}'$ be the node distributions obtained via t steps of random walk for two input images $\mathbf{x}$ and $\mathbf{x}'$. Note that $\mathbf{r}$ can be viewed as a new semantic representation for image $\mathbf{x}$, encoding the relation of the image to the entire graph. Based on this intuition, we define the random walk distance to be $d^{RW}(\mathbf{x}, \mathbf{x}') = d^{\chi^2}(\mathbf{r}, \mathbf{r}')$, which measures the χ^2 distance between the two images in this semantic space.

6 Discussion of Computational Costs

In this section we discuss the computational costs of our approach and possible strategies to reduce them. We factor out from this discussion the creation of the graph, since this is done only once during an offline stage and it can be reused for all inputs, regardless of their class. The graph is represented as a sparse matrix which occupies little space in memory. At test time the most expensive operation is computing the visual distances from the input to all nodes. Without any optimization, this operation takes about 2 minutes per input. However, this runtime can be greatly reduced by adopting efficient NN search methods: for example, the system in [27] runs in a couple of seconds on a database of 10M images by using product quantization on classeme vectors to speed up the search with little loss in accuracy. The RW embedding takes on average 99 seconds per input when using $t = 25$ steps, but also this operation could be made much faster by reformulating the walk in terms of powers of eigenvectors of the matrix A, as discussed in [25]. The SEO optimization on average runs in 48 seconds per example on a standard budget PC using $m = 400, n = 100, l = 4$.

7 Experiments

We now describe experimental evaluations of our distance metrics on the following datasets: Caltech101, Caltech256 [28] and the ILSVRC2010 database [29]. Unless otherwise noted, all results are based on a graph constructed from the 10M ImageNet dataset using connectivity $k = 10$ (see [30] for our study of the sensitivity to the size of the auxiliary dataset N and the graph connectivity k).

7.1 Evaluation of metrics for "same or different class" recognition

We begin by presenting results on the Caltech101 dataset. While this image database is known to be simple for the recognition standards of modern categorization systems, it was the dataset used in [11] to compare different image metrics. We follow the experimental setup used in [11]: we use the same set of 1020 photos (10 samples for each of the 102 classes); the set is split in two subsets of 51 classes; each subset is used in turn as training and testing set, so as to tune the parameters with two-fold cross validation. The final result is presented as the average cross-validation error. In each cross validation set, there are 129,795 distinct image pairs: in 2295 of these pairs the two images contain an object of the same class, while in the remaining pairs the two images belong to different category. In this experiment the value of distance is directly used to make a classification decision on whether the two samples contain the same object. As in [11], the result is presented in terms of Area Under the Curve (AUC) computed from the ROC curve.

We consider in our evaluation our two proposed embedding methods – **SEO** and **RW**. In addition, we include the simple embedding obtained by connecting each test image to its n closest visual neighbors in the graph, and denote

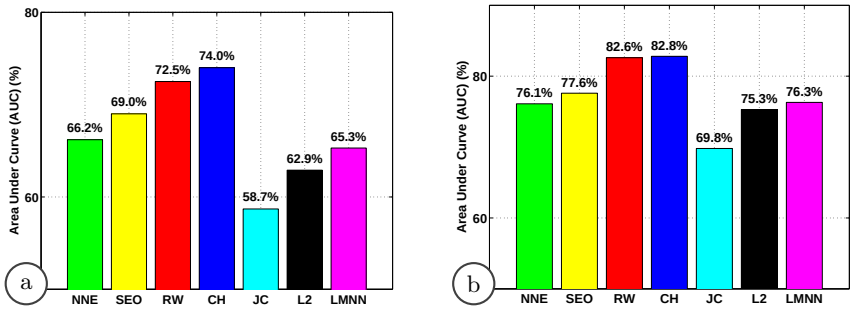

Fig. 2. Performance of distance metrics on Caltech101 using (a) GIST and (b) classeme features. Our metrics based on embedding in the graph are: NNE, SEO, RW. The ImageNet metrics proposed in [11] are CH and JC. The visual distances are L2 (the Euclidean metric) and LMNN (learned using the method of [16]).

this embedding method as Nearest-Neighbor Embedding (**NNE**). For NNE and SEO, the final semantic distance is computed as the shortest path distance between the two embedded nodes. As discussed in section 5, for the RW embedding we compute the semantic metric as the χ^2 distance between the random walk probability vectors. In addition to these metrics, we include the two distances proposed in [11]: **CH** is the "ImageNet" category histogram metric, while **JC** is the distance inspired by the Jiang-Conrath semantic similarity [31]. All parameters were optimized individually for each method by considering the following values: $m \in \{100, 200, 400, 800\}, n \in \{5, 10, 100, 200\}, l \in \{1, 2, 4, 6\}$. We also present results for two baseline metrics that do not use the auxiliary ImageNet database: **L2** denotes the L2 distance between the feature vectors of the two input images; **LMNN** indicates the distance learned using the large-margin nearest-neighbor approach described in [16]. Even for LMNN, we trained and tested the metric by using two-fold cross validation (i.e, the training and test sets involve two sets of disjoint classes), with 10 samples per class. To train this metric we used the software provided by the authors and as recommended in the manual we preprocessed the feature vectors via PCA, tuning the PCA target dimensionality for the best possible accuracy.

The performances of the different metrics are shown in figure 2. We can see that SEO, RW and CH perform considerably better than the visual distances (L2 and LMNN), with RW and CH nearly tied as the best metrics. The use of the auxiliary labeled data enables these distances to infer additional semantic connections yielding large improvements over LMNN, which has been previously shown to be one of the best metric learning methods (see, e.g., evaluations in [16, 6, 11]). It is also interesting to notice that both SEO and RW perform much better than the naïve NNE strategy which directly links the test images to their visual neighbors: this suggests that the semantic coherence enforced by SEO and RW produces a beneficial refinement of the initial set of visual neighbors.

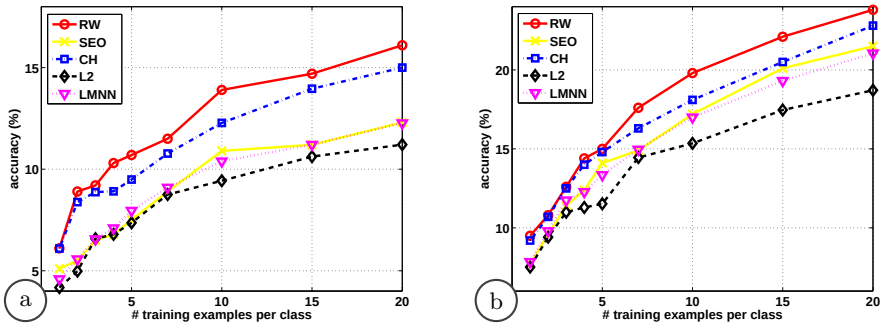

Fig. 3. Caltech256 multiclass recognition using a NN classifier based on different image metrics using (a) GIST and (b) classeme descriptors. Our RW metric gives consistently the best results: it even outperforms the LMNN metric, which in this experiment has been advantageously trained on the test categories.

7.2 Using semantic distances for multiclass object categorization

In this section we demonstrate the use of semantic distances to perform multiclass object recognition using two different classification models – the NN classifier and a SVM trained with kernels defined by our metrics.

Nearest-neighbor classification with semantic metrics. We begin by presenting an evaluation on the Caltech256 dataset. The test set was obtained by sampling 10 images from each of the 256 classes. The training set size is varied from a minimum of 1 to a maximum of 20 examples per class. We use the NN classifier to perform multiclass recognition as follows: for each test image, we compute its distance to the training examples of all 256 classes and then pick the class most voted among the K nearest neighbors, where K is an integer optimized individually for each distance metric. Note that the embedding of the training images in the graph is done without exploiting the textual tags of the Caltech256 classes. We report the NN classification accuracy obtained with the RW, SEO, CH, L2 and LMNN metrics (we omit NNE and JC as they produce much poorer results). Here the LMNN metric was learned from a separate training set of 10 images for each of the 256 classes: thus the LMNN method here is given the significant advantage of training on the test classes. Figure 3 shows the recognition accuracy as a function of the number of training examples for (a) GIST and (b) classeme features. We see that on this task RW outperforms all distances, including CH as well as the LMNN metric trained in highly favorable conditions. The SEO metric performs better than the L2 distance but not as well as the RW and CH metrics.

We now describe NN multiclass recognition on a subset of the ILSVRC2010 dataset. This is a difficult test: the ILSVRC2010 images are more challenging than those in the Caltech sets as they often contain multiple objects, and exhibit a much wider within-class variance. However, the downside of this test is that the

ILSVRC2010 classes are present as synsets in the ImageNet dataset (although the two sets of images are, of course, disjoint). This means that the test categories are also included in the manifold. For this experiment, we sample 5 images from 500 randomly selected categories. We then partition the dataset into 5 subsets, each containing one example of each class. We use this partition to evaluate the 5-fold cross validation error of the NN classifier using different metrics. The recognition accuracies using classeme features are $\mathbf{6.04}\%, \mathbf{5.08}\%, \mathbf{2.92}\%$ for RW, CH, and L2, respectively. Note that while the absolute accuracy rates are low due to the small number of training examples per class (only 4 for each validation run) and the large number of classes, the RW metric provides a relative improvement of 18% over the accuracy obtained using the state-of-the-art CH distance.

Nonlinear SVM classification with semantic kernels. We conclude by presenting experiments demonstrating that our metrics can be used to construct powerful kernels for nonlinear Support Vector Machines (SVM). We compare kernels built from our distances to popular hand-defined kernels for categorization and show that in all cases our RW metric provides superior results.

Most kernels for SVMs are defined so that the kernel distance is close to 1 when the input vectors are similar and near to 0 when the inputs are highly different. In order to achieve this desired behavior with our metrics, we apply the "exp" function to the negative values of the distances, i.e., we define the kernel as $k(\mathbf{x}, \mathbf{x}') = \exp(-d(\mathbf{x}, \mathbf{x}')/\gamma)$, where d is the semantic distance and γ is a hyperparameter. We denote with expRW, expSEO, and expCH the kernels built by using as distance d the metrics RW, SEO and CH, respectively. Note that expRW and expCH are obtained by applying the exponential function to negative χ^2-distances, which always yields a Mercer kernel [32]. Instead, expSEO may produce a kernel matrix that is not Mercer. When this happens, we follow the common practice of thresholding the negative eigenvalues of the distance matrix to zero in order to yield a proper kernel matrix [33]. Finally, we include as baselines the exponential kernel (expL2) and the Gaussian kernel (Gaussian), both built by applying the exponential function to distances between visual descriptors: $k^{\mathrm{expL2}}(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||/\gamma)$ and $k^{\mathrm{Gaussian}}(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2/\gamma)$. These two kernels are commonly used for image classification.

We evaluate this set of kernels on the Caltech256 dataset, using 15 training examples per class. With the Gram matrix of each kernel, we train a nonlinear one-vs-the-rest SVM by optimizing the dual objective. The SVM regularization parameter C and the kernel hyperparameter γ are selected individually for each method via 5-fold cross validation. We evaluate the resulting SVMs on a test set of 10 images per class as in the previous subsection. We include in this comparison also the dot-product kernel $k^{\mathrm{linear}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$, which produces a linear SVM. The results are shown in fig. 4 for both (a) GIST and (b) classeme features. From this plot we see that the kernels defined by our RW distance yield much higher accuracy than the traditional hand-defined nonlinear kernels considered here. As in all our previous experiments, even in this evaluation our RW metric matches or outperforms the CH distance proposed in [11].

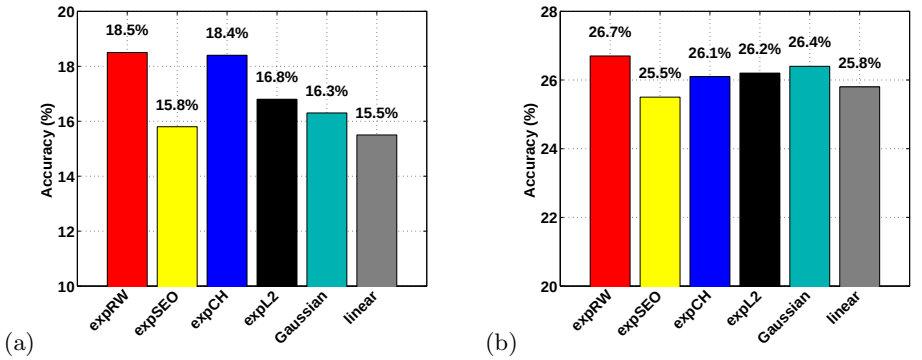

Fig. 4. Caltech256 performance of nonlinear SVMs trained with different kernels using (a) GIST and (b) classeme features: expRW and expSEO denote kernels constructed from our RW and SEO distances; expCH is the kernel induced by the CH distance of Deselaers and Ferrari [11]; exp-L2 and Gaussian are the exponential and Gaussian kernels computed from the L2 visual distances; linear indicates the linear SVM learned using the dot-product kernel. The training set consists of 15 examples per class.

8 Conclusions

We have presented new image metrics for categorization. Our distances are computed by embedding the photos in a semantic image manifold. This allows our methods to infer semantic relations that cannot be captured by directly comparing the two input images. We have shown that this yields results matching or outperforming the state-of-the-art on three different datasets. Our current embedding methods require calculating distances to all nodes in the graph. To reduce this cost in the future we are interested in learning parametric embedding models. Our graphs and image embedding software may be obtained from [30].

Acknowledgments. We are grateful to S. Nowozin and C. Rother for useful discussion on strategies to optimize our SEO energy and to T. Deselaers and V. Ferrari for sharing data. Thanks to A. Bergamo for help with the experiments.

References

1. Carey, S., Bartlett, E.: Acquiring a single new word. In: the Stanford Child Language Conference. (1978)
2. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV. (2007)
3. Wang, G., Hoiem, D., Forsyth, D.: Learning image similarity from flickr using stochastic intersection kernel machines. In: Intl. Conf. Computer Vision. (2009)
4. Malisiewicz, T., Efros, A.A.: Recognition by association via learning per-exemplar distances. In: CVPR. (2008)
5. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR. (2009)

6. Ramanan, D., Baker, S.: Local distance functions: A taxonomy, new algorithms, and an evaluation. IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 794–806
7. Tenenbaum, J.B., Kemp, C., Griffiths, T.L., Goodman, N.D.: How to Grow a Mind: Statistics, Structure, and Abstraction. Science **331** (2011) 1279–1285
8. Tenenbaum, J.B., Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science **290** (2000) 2319–2323
9. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. JMLR (2006)
10. Rosch, E.: Cognitive representations of semantic categories. J. of Experimental Psychology: General (1975)
11. Deselaers, T., Ferrari, V.: Visual and semantic similarity in ImageNet. In: CVPR. (2011) 1777–1784
12. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans. PAMI (2008)
13. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009) 248–255
14. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighbourhood components analysis. In: NIPS. (2004)
15. Torresani, L., Lee, K.C.: Large margin component analysis. In: NIPS. (2006)
16. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research **10** (2009) 207–244
17. Babenko, B., Branson, S., Belongie, S.: Similarity metrics for categorization: From monolithic to category specific. In: ICCV. (2009) 293–300
18. Hastie, T., Tibshirani, R.: Discriminant adaptive nearest neighbor classification. IEEE Trans. PAMI (1996)
19. Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest-neighbor classification. IEEE Trans. Pattern Anal. Mach. Intell. **24** (2002) 1281–1285
20. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Visual Perception, Progress in Brain Research **155** (2006)
21. Torresani, L., Szummer, Fitzgibbon, A.: Efficient object category recognition using classemes. In: ECCV. (2010)
22. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: WordNet: An on-line lexical database. International Journal of Lexicography **3** (1990) 235–244
23. Lim, Y., Jung, K., Kohli, P.: Energy minimization under constraints on label counts. In: ECCV. (2010)
24. `https://http://www.gurobi.com/`.
25. Craswell, N., Szummer, M.: Random walks on the click graph. In: SIGIR. (2007)
26. Szummer, M., Jaakkola, T.: Partially labeled classification with markov random walks. In: NIPS. (2001)
27. Rastegari, M., Fang, C., Torresani, L.: Scalable object-class retrieval with approximate and top-k ranking. In: ICCV. (2011) 2659–2666
28. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, Caltech (2007)
29. Berg, A., Deng, J., Fei-Fei, L.: Large scale visual recognition challenge (2010) http://www.image-net.org/challenges/LSVRC/2010/.
30. `http://vlg.cs.dartmouth.edu/semanticembedding`.
31. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. CoRR (1997)
32. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
33. Duchenne, O., Joulin, A., Ponce, J.: A graph-matching kernel for object categorization. In: ICCV. (2011) 1792–1799