Hierarchical Classification of Images with Sparse Approximation

Byungsoo Kim, Jae Young Park, Anna C. Gilbert, Silvio Savarese

University of Michigan, Ann Arbor
bsookim,jaeypark,annacg@umich.edu, silvio@eecs.umich.edu

Abstract. Using image hierarchies for visual categorization has shown to have a number of important benefits. For instance it enables a significant gain in efficiency (e.g., logarithmic with the number of categories [1, 2]). Moreover, arranging visual data in a hierarchical structure echoes the way how humans organize data and enables the construction of a more meaningful distance metric for image classification [3] (see figure 1). However, a critical question still remains controversial: would structuring data in a hierarchical sense also help classification accuracy? While our intuition suggests that the answer may be positive, up to date no method have shown conclusive results that can demonstrate the correctness of this claim for the most general case of large scale databases. In this paper we address this question and show that the hierarchical structure of a database can be indeed successfully used to enhance classification accuracy using a sparse approximation framework. We propose a new formulation for sparse approximation problem where the goal is to discover the sparsest path within the hierarchical data structure that best represents the query object. Extensive quantitative and qualitative experimental evaluation on a number of branches of the Imagenet database [4] as well as on the Caltech 256 [2] demonstrate our theoretical claims and show that our approach produces the best categorization results (in term of a number of hierarchical-based distance functions) over a number of competing large scale classification schemes that do not exploit the hierarchical structure of the database.

1 Introduction

Recent advances in computer vision and image-based search have enabled the design of recognition methods that are capable to classify images into large number of visual categories (typically, hundreds) [5–8]. In a current paradigm for image categorization, image classes are organized in a flat structure and the problem is the one of discovering the class (among all those in the flat structure) that best represents (in term of a distance function) the visual content of a given query image. The classification error is measured by inspecting whether the recognized class label is equal to the ground truth one or not. While this classification paradigm has shown promising classification results, a number of questions remain unanswered: i) how well classification scales as the number of

categories increases? Typical schemes lead to linear or even quadratic complexity [9, 10]; ii) is this the correct metric (distance function) to measure a classification error? We can argue that having a dog been misclassified as a stapler is "worse" than having a dog misclassified as a cat (Fig. 1); iii) is this the "natural" way to organize visual data for classification? iv) would a different structure (than flat) help improve the classification accuracy?

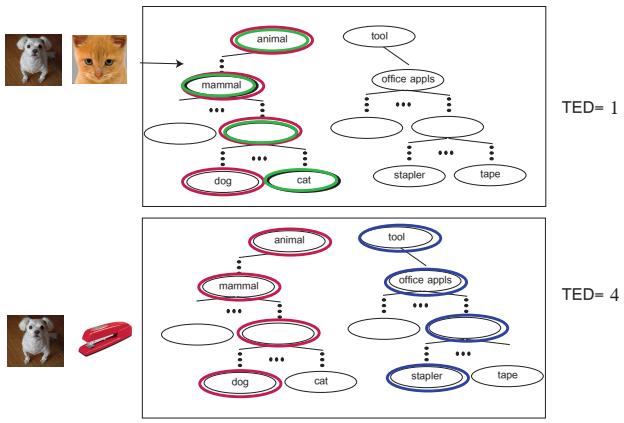

Fig. 1. The similarity between two objects can be measured as a distance between two paths in the trees - such distance is measured as the TED as discussed in Sec. 4.

An emerging paradigm has advocated the idea of organizing visual data in a hierarchical structure rather than in a flat one. This paradigm addresses some of the questions raised above: i) it enables a significant gain in efficiency, typically logarithmic with the number of categories [1, 2]; ii) it enables the construction of a more meaningful distance metric for image classification ; iii) arranging visual data in a hierarchical structure echoes the way how humans organize data [3]. However, a critical question still remains controversial: would structuring data in hierarchical sense also help classification accuracy? Up to date there is no definite answer to that question. For instance, top-down classification schemes (applied on hierarchical structures) such as [1, 2] have produced inconclusive evidence as for whether hierarchy has a beneficial effect on classification accuracy. Classification methods based on Hierarchical Support Vector Machines can be used to trade off accuracy against speed [2]. Methods based on combining models from different levels of the hierarchy [11] have shown some positive signal but some of the assumptions (related to the precision/recall rate associated to child/parents) are not verified and the hierarchical structure should be deeper and larger than the one tested in [11] (which comprises a handful of categories only).

In this paper we attempt to address the issues discussed above and show that the hierarchical structure of a database can be successfully used to enhance classification accuracy using a sparse approximation framework. The key idea is to introduce a distance function that takes into account the hierarchical structure of the visual categories (Fig. 1) and identify two images to be similar if they share a similar path in the hierarchy. We show that this distance function (or similarity metric) is equivalent to the Tree Edit Distance (TED). This allows to cast the categorization problem as the one of discovering the category in the tree structure that has the smallest TED from the query category label. We solve this problem via sparse approximation and introduce a new formulation of the sparse approximation problem which we call *hierarchical* sparse approximation. In the typical sparse approximation problems, [12–14], a query image can be identified as the sparsest representation over the set of training images for all object classes; that is, the sparsest solution is one (or a combination of a few) image out of *all possible* images in the dataset. We call this the *flat* sparse approximation problem. The key novelty of our approach relies on the idea of that the sparse representation is not constructed over a *flat* structure of object classes (as in the classic sparse sensing problem) but rather by enforcing that the solution must be one (or a combination of a few) path out of all possible paths on a given hierarchy of object classes (training set).

Since our method relies on the sparsity of the representation, our approach is suitable for large scale classification problems; i.e., the conditions underlying the sparsity assumptions are best verified when the dataset is large and distribution of visual categories is diversified. In this work we present sufficient conditions under which our hierarchical sparse formulation can be used with success and small error bounds are guaranteed. Furthermore, a crucial property of our classification framework is that it is capable to classify multiple object instances at the *same time* if more than one (dominant) object appears in the query image (Fig. 2).

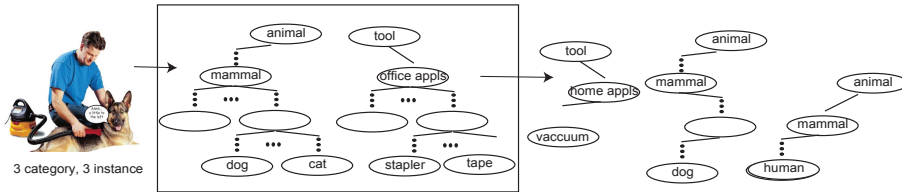

Fig. 2. Multi-instance classification. Given an image that contains multiple object instances from different categories, our algorithm is capable to discover the path associated to each of these objects.

We have carried out extensive quantitative and qualitative experimental evaluation on a number of branches of the Imagenet database [4]. Each branch

comprises hundreds of visual categories organized in the hierarchical structure. All the experiments demonstrate that our hierarchical sparse sensing framework yields much better classification accuracy over a number of benchmark classifications schemes: i) *flat* sparse approximation; ii) *flat* sparse approximation modified according [12]; iii) Support Vector Machines; iv) Hierarchical Support Vector Machines. Evaluation was carried out by comparing average precision measured in term of TED as well as by measuring the actual classification accuracy at each level of the hierarchy. Our method achieves a performance increase ranging from 10% to 40% for the most critical levels of the hierarchy. Additional experiments on multi-instance classification also show very promising results.

The rest of this paper is organized as follows. In Section 3, we will briefly review how sparse approximation can be used for image classification problem. The formal definition of hierarchical classification and embedding scheme is provided in the Section 4. Number of experiments are performed to validate our scheme in the Section 5. Finally, we summarize our proposal in the Section 6.

2 Related Work

The usage and construction of hierarchies for object categorization has received substantial interest in the vision and machine learning community. While the contributions discussed above [11], [15], [2], [1] focussed on using hierarchical structures to help or speed up classification, several other works proposed tools for learning of hierarchies of visual categories (i.e. constructing taxonomies) in unsupervised or semi-supervised fashion [16], [15], [2] as well as for exploiting multi-level structures of classifiers [17], [18]. Researchers also presented methods for organizing low (i.e., feature or part) level object representations organized in hierarchical fashion so as to increase descriptiveness and discrimination power [19], [20].

Sparse approximation for an image classification was first introduced by [12] for face recognition. Authors [12] framed the face classification problem as the one of finding a query class among large number of candidates (classes are organized in a flat structure) using sparse approximation and demonstrated that their framework is robust to noise and can handle occlusions. Recently, concepts from sparse coding were introduced to construct more descriptive dictionaries to represent visual categories [21].

3 Image Classification using Sparse Approximation

In this section, we describe our image representation and the introduce the basic formulation of the flat image classification problem based on sparse approximation. We assume a database of images is available. Further we assume that such database comprises a large number of categories and each category has large number of images of object instances. We assume that each image has a dominant object instance as in Caltech-101, Caltech-256 or the ImageNet [4]. In classification, we assume that the query image (with unknown category label) contains an object whose category label is included in the dataset. Of course, the query object itself is not included in the dataset. The classification problem can

be solved by seeking, among all the images (object instances) in the database, the one that is closest to the query object. The class such image belongs to is the classification result.

Object representation. Assessing whether an image is close to another relies on the construction of a distance function, and such distance function relies on the way we represent objects and the visual content of an image. Following a common representation used in the computer vision community, we describe an image using a normalized histogram of codewords (bag of words representation) [7] or, equivalently, a histogram capturing a spatial pyramid of codewords [8, 9]. In either cases, we denote such histogram by vector a x. Codewords are drawn from a learnt dictionary of vector quantized features as described in [7–9]. The size of the dictionary is denoted by K. Thus x is a column vector of size K, if we use a basic bag of words representation as in [7]. Notice that other type of representations are also possible.

Distance function. The similarity between two images represented by x_i and x_j can measured by computing the l_n norm distance between x_i and x_j, where n can be 0, 1, etc. Thus, similar images will have a small distance function.

Model matrix. Let us stack all the images in the database in a matrix H. Columns of H will correspond to column vectors x. Thus, H will be $K \times N$, where N is the number of images in the dataset. We call this matrix H the *model matrix*. Any query image can be then interpreted as a superimposition of one or a few images in the training data. That is, a query image can be expressed as

$$x = Hm, \tag{1}$$

where m is an indicator function that is 1 in correspondence of the images in the database that contribute to represent the query image by superposition. m will be zero otherwise. m is called the *mixing matrix* and is $1 \times N$ vector. A similar representation was introduced in [12] in the context of face recognition. Under the assumption that the database is large, the number of images that contribute to the construction of any query image will be extremely small, hence m is highly sparse as most the its entries will be zero.

Classification. Clearly m contains the information that allows us to estimate the class label of the query image. Therefore, the classification problem (*what is the object class?*) is recast into the problem of estimating the vector m (*where is a nonzero entry?*). Intuitively this formulation allows us to discover multiple dominant object instances in the image. Suppose the image contains three objects as in the Fig. 2. Then this query image can be expressed as a superposition of $K = 3$ training histograms (one may come from the dog class, one from human class, and one from vaccum class). Hence, the nonzero entries of m will return the 3 classes appearing in x. Note that the sparsity assumption still holds as long as the number of objects appearing in the query image is small. (We will study this condition in details later.)

Solving the Equation 1 is challenging because the system is underdetermined and has infinite number of solutions (note that we assume that the matrix H

has full rank). Because we postulate or seek a sparse mixing vector m, we can formulate this problem as a sparse approximation problem and seek to find the sparsest solution that best approximates (in ℓ_2 error) the observed instance. (The pseudo-norm $\|\cdot\|_0$ counts the number of non-zero entries in a vector.)

Problem 1

$$\min \|m\|_0 \quad \textit{subject to} \quad \|Hm - x\|_2 \leq \epsilon.$$

Unfortunately, the above problem is an NP-hard problem in general (given an arbitrary matrix H). We can, however, solve this problem in polynomial time with appropriate geometric assumptions on the matrix H. Let us assume for now that the training set containes the query image x. As proposed by [22, 12], one method is to observe that Equation 1 is an optimization problem with a non-convex objective function and that a convex relaxation of this problem yields a problem which can be solved efficiently with standard optimization techniques [13],

Problem 2

$$\min \|\|m\|_1 \quad \textit{subject to} \quad \|Hm - x\|_2 \leq \epsilon.$$

A second algorithmic approach is to use a greedy algorithm, one that identifies image instances iteratively, such as Orthogonal Matching Pursuit (OMP). See [14] and the references therein for details on this algorithm. Both algorithmic approaches are valid under the same geometric conditions on H. Because we use different flavors of a greedy algorithm, we focus on those results only. In order to interpret these algorithmic results, we must first define an important geometric quantity, the coherence of the database $\mu(H)$.

Definition 1 *The coherence $\mu(H)$ of the learned database is the maximum inner product between distinct histograms*

$$\mu(H) = \max_{i \neq j} |\langle H_i, H_j \rangle|.$$

Suppose that our observed instance x consists of k instances in the learned database H and let Λ be the index set of those instances. The following theorem provides us a geometric constraint on H and the similar set Λ, the Exact Recovery Condition, that guarantees OMP will recover the similar set.

Theorem 1 (ERC). *A sufficient condition for* OMP *to identify Λ after k steps is that*

$$\max_{\ell \notin \Lambda} \|H_\Lambda^+ H_\ell\|_1 < 1$$

where $H^+ = (H^* H)^{-1} H^*$ [14].

We can guarantee that the ERC holds as long as the

Theorem 2. *The ERC holds whenever $k < \frac{1}{2}(\mu^{-1} + 1)$. Therefore,* OMP *can recover any sufficiently sparse signals [14].*

If, instead of exact replication of learned instances, the observed instance consists of only very similar instances in the learned set H, i.e. training set does not contain the query image, we cannot hope to exactly recover x. Instead, we aim to find a mixing vector m with k non-zero entries that gives us the closest approximation to x. Let us call that mixing vector m_k and the smallest approximation error $\|x - Hm_k\|_2 = \|x - x_k\|_2$. If we do not seek to identify too many similar objects, then OMP will find k instances that are close to the most similar ones.

Theorem 3. *Assume* $k \leq \frac{1}{3\mu(H)}$. *For any observed instance* x, *the approximation* $H\widehat{m} = \widehat{x}$ *after* k *steps of* OMP *satisfies*

$$\|x - \widehat{x}\|_2 \leq \sqrt{1 + 6k}\|x - x_k\|_2$$

Note that we do not get a guarantee on the quality of the recovered m vector. We do know that $\|H\widehat{m} - x\|_2$ is close to the best k-term approximation. This should be contrasted with a guarantee that $\widehat{m}$ is close to m_k.

4 Hierarchical Classification with Sparse Approximation

In this section, we investigate the relationship between hierarchical classification and the sparse approximation. We start with the theoretical argument that a small error in the mixing vector $\|\widehat{m} - m\|\|_2$ or in the reconstruction of the observation x does not guarantee hierarchical similarity between $\widehat{m}$ and m. This problem is depicted in a toy example in the Figure 2. In the figure, the ground truth label is the dog. Consider two estimation stapler and cat. Given the hierarchical structure of the database, we can say that the sparse approximation by cat is better than the approximation by stapler. The errors in the mixing vector, however, are the same. This means that our method of assessing the quality of the approximation does not take into account the hierarchical structure of the database. In order to obtain a more appropriate sparse approximation, we observe that the learned database is organized in a (rooted, labeled, recursive) tree and that we can group instances in the same category into nodes of the tree. See Figure 1 for an illustration. Furthermore, the tree structure induces a different distance metric than that implied by the previous discussion of sparse approximation. Our previous model assumes a flat index structure of the object instances; the histograms for the objects are numbered 1 to N and concatenated as column vectors into the matrix H. This indexing structure does not reflect the distance between nodes in the tree—all instances are equally far apart, regardless of the node to which they are associated. To exploit this structure, we introduce an embedding of the tree structure into the flat index scheme that more closely captures the hierarchical structure.

4.1 Hierarchical Embedding

To embed the structure of the database in our indexing scheme, first we convert individual isolated nodes into paths in the tree by augmenting the definition of a representation. Let I denote the set of object instances in the observed image.

For each object $i \in I$, we create a path p_i from i to the root of the tree and we let P be the union of all paths p_i. Next, we observe that two different sparse representations with two different sets I_1 and I_2 of object instances give rise to two different sets of paths P_1 and P_2. As P_1 and P_2 are themselves trees, a natural way to measure the distance between them and one that incorporates our intuition that objects from "nearby" nodes are more similar than those that are "far away" is *tree edit distance* (TED) [23].

There are three possible tree edit operations: rename, delete, and insert. For formal definition of TED, assume that we are given a cost function defined on each edit operation. An edit script S between two trees P_1 and P_2 is a sequence of edit operations that turn P_1 into P_2. The cost of the script S is the sum of the costs of the operations. An optimal edit script between the two trees is an edit script of minimum cost and we define this cost as the tree edit distance.

To capture the notion of small tree edit distance in the mixing vector, given an instance set I, we form the set of paths P from each object instance to the root (counting multiplicities of the paths) and encode the characteristic function of the (multiset) P in a new vector l. That is, $l(j)$ counts the number of paths from some object instance $i \in I$ to the root that pass through node j.

We claim that the ℓ_2 distance between two such embedded vectors l_1 and l_2 is close to the TED of their respective augmented trees. Furthermore, for a given instance set I, we can obtain the vector l from the mixing vector m associated to I in a linear fashion by multiplying by a sparse matrix E (i.e., $l = Em$). A second useful property of the vector l is that if m is sparse, then l remains fairly sparse; the increase in the number of non-zero entries is no more than a factor D where D is the maximum depth of the tree. Therefore, we can solve an embedded sparse approximation problem that more accurately reflects the hierarchical structure of the database.

Problem 3

$$\min \|l\|_0 \quad \textit{subject to} \quad \|\tilde{H}l - x\|_2 \leq \epsilon,$$

where $\tilde{H} = HF$ and $l = Em$. Here, F is a matrix such that $FE = Id$ holds. A complete optimization of both the embedding matrix E and its left-inverse F is beyond the scope of this paper. We use $F = E^+$ for our purposes. Problem 3 can be solved efficiently by a greedy algorithm referred to as Tree-OMP [24]. Tree-OMP is similar to the greedy algorithms we discussed previously with the additional step that for all non-zero components in the vector l, the algorithm assumes that all the components that correspond to ancestors in the tree are non-zero as well and computes their values (instead of assuming they are zero as in OMP). Unlike OMP, no theoretical analysis of this algorithm exists; it does, however, perform well in practice on hierarchical data as we shall see next.

4.2 Sparse Path Selection Algorithm

After the vector l is returned by solving problem 3, we obtain an estimate of the path in the hierarchical database associated to the query image. We perform a post processing step which we call Sparse Path Selection (SPS). The reason

for doing this is the following. Ideally, the sparsest solution of problem 3 should return a vector of "1" and "0" where the non-zero elements in l allows to estimate the category labels of the query object as well as its parents. Unfortunately, this not always the case and values between "0" and "1" can also found because of the estimation noise. To solve this issue, we introduce a threshold and interpret as a positive response any value that is above such threshold (and as negative response, otherwise). Finding this threshold, however, is not trivial as it may be different if different datasets are used. Thus, in our experiments, we propose to automatically learn this thresholds using a binary MAP estimator trained using a validation set. Such evaluation set is then removed from the dataset so as to avoid contamination during testing. Our classification scheme can be summarized as follows:

Algorithm 1 *Sparse Path Selection Algorithm* SPS

1. Input: form the matrix H of training vectors collected from all images in the dataset

2. Encoding: $l = Em$, $\tilde{H} = HF$, where $F = E^{+}$

3. Normalize the columns of $\tilde{H}$ to have unit l^2-norm.

4. Solve the hierarchical sparse approximation problem 3 using Tree-OMP *and estimate l*

5. Truncate noise by learned thresholds value and return classification results.

5 Experiments

In this section, we present quantitative and qualitative experimental results to validate our theoretical claims. We test our algorithm using different hierarchical databases. These are: i) 3 branches of the ImageNet [4] each comprising hundreds of categories; ii) The hierarchical Caltech-256 dataset [2]. We use different metrics to evaluate the performances of our algorithm: i) Overall average Tree Edit Distance (TED); ii) Average classification accuracy for each levels of the hierarchy; iii) Overall average classification accuracy. We benchmark our results using two state-of-the art large scale classification methods. These are: i) SRC: the sparse approximation technique introduced by [12]; ii) Pyramid Matching SVM [8].

In each of these experiments we used 16 grid patches with spacing of 8 pixels to generate SIFT descriptors. BoW histograms are constructed using 500 codewords generated from K-means clustering. Finally, we used SPH (Spatial Pyramid Histogram) up to the resolution level 4 to represent each image.

In each experiment we sample (at most) 100 images for each node of the working database and use these for learning. (E.g. to build the H matrix) As an example, for the *domesticAnimal* tree of ImageNet we collected about 21000 images for training. We sample an additional 10 images per node for testing. This way testing images are guaranteed to be different from those in the training set. In all these experiments, we tested the case where a query image contains only a single category. We show anecdotal examples of multi-instance classification in the last section.

ImageNet Subsets ImageNet [4] is a hierarchical image database with $10,000,000$ images across over $10,000$ categories. It organizes the different classes

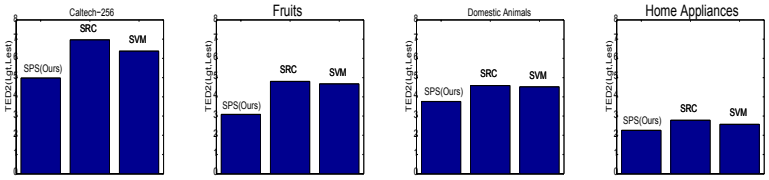

Fig. 3. Average Tree Edit Distance (TED) for different subcategories is drawn.

of images according to the WordNet [25] structure, and "IS-A" relationship exists between parents and children. In the experiments, we used 3 different branches from the ImageNet: *Home Appliances, Domestic Animals, and Fruits*. These subsets are chosen to see the performance changes according to differences in the number of classes (48, 212, 320, respectively). Also, they have different hierarchical structure: *Domestic animals* has a deep hierarchy and *Home Appliances* has relatively small hierarchy. Moreover, we created a new version of *Domestic animals* where images only reside in the leave nodes, and no images are in the internal nodes.

Hierarchical Caltech-256 The Caltech-256 is rearranged in a hierarchy according to best matches in the WordNet. In this Hierarchical Caltech-256, every categories lies on the leave of the hierarchy and internal nodes have no images.

Benchmarks We use use two state-of-the-art methods for large scale classification: 1) the sparse approximation technique introduced by [12] (SRC). We use problem 1 (Sec.3) to find the solution m via sparse approximation (similarly to [12]). We use the post-processing procedure in [12] to estimate the final class label. Notice that this method does not exploit the hierarchical structure of the database and "sees" the database as flat. Notice that SRC returns a single class label (not a path in the tree) which can be used to form the mixing matrix m_{SRC}. In order to compare SRC results with ours, we transform m into its corresponding path $l_{SRC} = m_{SRC}$; 2) the SVM Pyramid Matching classification scheme ([8]). Given a query image, we use this technique to return the class label by SVM classification using all the categories in the training tree as a model. This approach was also tested by [2] for Caltech 256. Similarly to the SRC case, the output m_{SVM} can be converted to its corresponding path using $l_{SVM} = Em_{SVM}$.

Hierarchical Similarity Verification

In this part, we give empirical results of hierarchical similarity in terms of TED (which is a natural distance function to compare the similarity of two paths in a tree). Thus, if ground truth path and the estimated path are similar, the TED will be small. In Fig. 3 we show average TED between ground truth paths and estimated path for all our testing images using our approach (SPS). In the same figure we also report the TED distance between ground truth path and path estimated by both SRC and SVM (i.e., l_{SRC} and l_{SVM} as discussed above).

Note that the TED associated to our approach is systematically smaller than that of SRC or SVM for all the datasets. This result supports our argument that the proposed framework actually guarantees small TED bounds. Notice that this bound is not guaranteed in the original flat sparse approximation formulation (i.e., without encoding scheme). Also, notice that when the hierarchical structure is relatively flat, the effect of encoding and the advantage from our framework becomes less significant.

Effect on Different Hierarchicy Levels TED returns a global measurement of path similarity regardless of the level and position in the tree. In this experiment we explore the performance of our framework at different levels of the tree. In the Fig. 4, the accuracy versus the levels of the hierarchy is drawn for different datasets. The plot reports the average number of correctly estimated nodes (categories) for each level (x-axis) for all testing image. A node j is estimated correctly if the ground truth path evaluated at j is equal to the estimated path at j for a given test image. Clearly, the root node is always classified correctly. As we go down toward the bottom of the tree, the likelihood of classifying nodes correctly becomes smaller and smaller. Note that this graph is always monotonically decreasing because whenever the estimation of the child category is correct, then parent category estimation is correct too. When the hierarchical level is low, the performance of our SPS is similar to SRC and SVM. However, when the hierarchical level increases the gap between our SPS and SRC or SVM become larger.

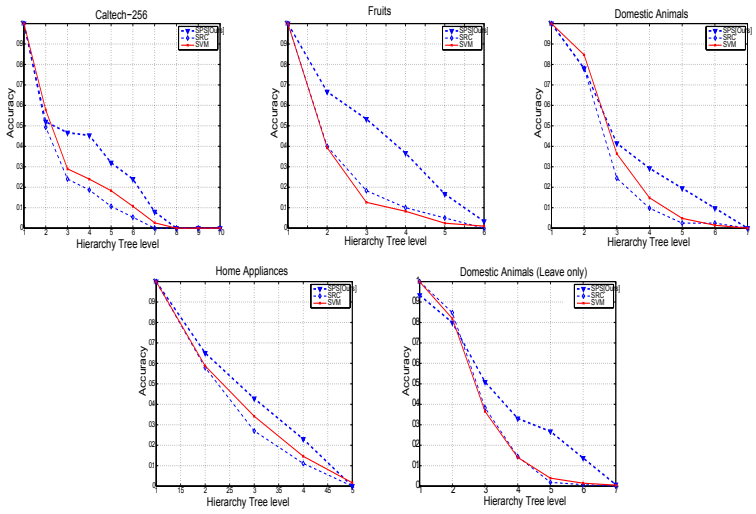

Fig. 4. Average accuracy of classification for different hierarchical levels. We tested on five different categories, Caltech-256, Fruits, Domestic animals, Home applications and Domestic animals (leaves only)

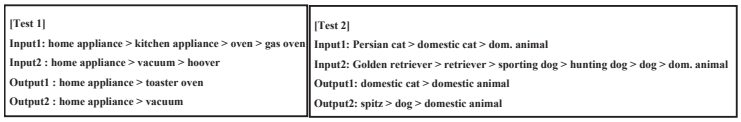

Fig. 5. Example of multiple object instance recognition.

In the Table 1, the average accuracy is shown as a function of number of nodes in each datasets. It is clear that in average, our proposal outperforms the competing classification schemes (see text for details).

Table 1. Average Accuracy

Algorithm	Fruits	D-Animals	D-Animals(Leave)	Home App	Caltech-256
SPS	0.46	0.40	0.43	0.46	0.31
SRC	0.28	0.31	0.34	0.39	0.21
SVM	0.27	0.35	0.34	0.42	0.24

Contribution from Internal and Leave Categories In this section, we investigate the contribution of internal nodes and their impact to classification accuracy. For this purpose, we generated a dataset called "Domestic Animals-Leave Only", which has the same hierarchy to the original "Domestic Animals" category, but where no images are included in the internal nodes (thus, only leaf nodes contain images). This means that the model matrix H is learnt as well as the testing is performed using leaf nodes only. Likewise, SVM and SRC will be tested and trained on such nodes only. The accuracy curve for this dataset is shown in the Fig. 4. Note that the accuracy performance of both "Domestic Animals" and "Domestic Animals-Leave Only" have very similar performance. This suggests that our method still works well if the hierarchical dataset does not contain images in the internal nodes. This is an useful property in real applications, as it is easier to make such a hierarchical tree without internal nodes when we generate a hierarchy structure from a flat structure. For example, the Hierarchical Caltech-256 can be easily constructed under these assumptions.

Multiple Object Recognition As discussed in the introduction, in this section we present anecdotal examples showing that our framework is able to classify images containing multiple instances. In such examples the histogram representing the query image can be expressed as a superimposition of multiple object category histograms. So, as discussed in the technical section, our SPS method will return *multiple* paths – a path for each of category in the query image. Examples in the figure 6 show some successful cases. Paths are reported in text format. We plan to extensively evaluate this capability in future work.

Examples: Hierarchical Classification In this section, we present anecdotal examples showing some the paths returned by our SPS compared with

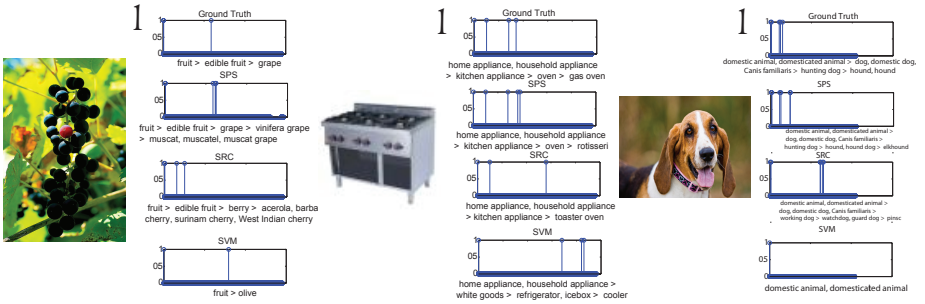

Fig. 6. The hierarchical path is estimated as nonzero entries in the encoded mixing vector l. Note that the path estimated by SPS (ours) is closer to ground truth path than SVM or SRC is.

those returned by SVM or SRC. Note that estimated parent nodes returned by SVM and SRC are much less accurate than those returned by SPS. Paths are reported in text format. See figure 6 and supplementary material.

6 Conclusion

In this work, we introduced a new framework for hierarchial classification using a new formulation of the sparse approximation problem. We demonstrated, for the first time (up to our knowledge), that the hierarchical structure of a large and complex database can be indeed successfully used to enhance classification accuracy. Experimental results on several large scale dataset were used to support our claims.

References

1. Marszalek, M., Schmid, C.: Semantic hierarchies for visual object recognition. (2007)
2. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. (2008) 1–8
3. Palmer, S.: Vision science: Photons to phenomenology. MIT press Cambridge, MA. (1999)
4. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proc. CVPR. (2009) 710–719
5. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. (2007)
6. Everingham, M., Zisserman, A., Williams, C., Van Gool, L.: The PASCAL visual object classes challenge 2006 (VOC2006) results. In: Workshop in ECCV06, May. Graz, Austria, Citeseer (2006)
7. Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV. Volume 1., Citeseer (2004) 22

8. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, Citeseer (2006)
9. Grauman, K., Darrell, T.: The pyramid match kernel: Discriminative classification with sets of image features, Citeseer (2005)
10. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision **57** (2004) 137–154
11. Zweig, A., Weinshall, D.: Exploiting object hierarchy: Combining models from different category levels. In: IEEE 11th International Conference on Computer Vision. (2007) 1–8
12. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009) 210–227
13. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM review **43** (2001) 129–159
14. Tropp, J.: Greed is good: Algorithmic results for sparse approximation. IEEE Transactions on Information Theory **50** (2004) 2231–2242
15. He, X., Zemel, R.: Latent topic random fields: Learning using a taxonomy of labels. In: IEEE Proc. Conf. on Computer Vision and Pattern Recognition (CVPR). (2008) 1–8
16. Bart, E., Porteous, I., Perona, P., Welling, M.: Unsupervised learning of visual taxonomies. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2008)
17. Amit, Y., Geman, D., Fan, X.: A coarse-to-fine strategy for multiclass shape detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2004) 1606–1621
18. Fan, X.: Efficient multiclass object detection by a hierarchy of classifiers. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. Volume 1. (2005)
19. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR. Volume 5., Citeseer (2006)
20. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR. Volume 3613., Citeseer (2007) 1575–1589
21. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Wood, M., Hengartner, N., Matzner-Lober, E., Rouvière, L., Burr, T., Malyshkina, N., et al.: Online learning for matrix factorization and sparse coding. stat **1050** (2009) 1
22. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Communications on Pure and Applied Mathematics **59** (2006) 1207
23. Bille, P.: A survey on tree edit distance and related problems. Theoretical computer science **337** (2005) 217–239
24. La, C., Do, M.: Tree-based orthogonal matching pursuit algorithm for signal reconstruction. In: IEEE International Conference on Image Processing (ICIP), (Citeseer) 1277–1280
25. Fellbaum, C., et al.: WordNet: An electronic lexical database. MIT press Cambridge, MA (1998)