Proceedings of the International Multiconference on Computer Science and Information Technology pp. 45–51
ISBN 978-83-60810-27-9
ISSN 1896-7094

Hierarchical Object Categorization with Automatic Feature Selection

Md. Saiful Islam, Andrzej Sluzek
Centre for Computational Intelligence, School of Computer Engineering
Nanyang Technological University, Singapore-639798
Email: saiful@pmail.ntu.edu.sg, assluzek@ntu.edu.sg

Abstract—**In this paper, we have introduced a hierarchical object categorization method with automatic feature selection. A hierarchy obtained by natural similarities and properties is learnt by automatically selected features at different levels. The categorization is a top-down process yielding multiple labels for a test object. We have tested out method and compared the experimental results with that of a nonhierarchical method. It is found that the hierarchical method improves recognition performance at the level of basic classes and reduces error at a higher level. This makes the proposed method plausible for different applications of computer vision including object categorization, semantic image retrieval, and automatic image annotation.**

I. Introduction

IN computer vision, object categorization is the problem of deciding the class of an object present in a given image. The major challenge in object categorization is the extraction of suitable features from the images with consideration of geometric and photometric transformations of an object in different image planes, intra class variations, object deformations, partial occlusions, and background cluttering etc. Different kinds of extractable features have been proposed over the last three decades [1]. It is generally agreed that a particular feature may not be suitable for all classes of objects. Some features may be specialized for discriminating between certain classes while being useless in the general cases. Hence, dividing the problem into sub-problems and selecting the appropriate subset of extractable features for each of them may be useful.

Different classes of objects could be clusterized by their natural superclass-subclass relationships and can be represented by a hierarchical (parent-child) structure. For example, taxonomy of biological objects is generally arranged in such a structure. In a hierarchy, a subclass by definition has the same properties and constraints as the superclass plus some additional properties or constraints. Human is capable to perceive such superclass-subclass relationship. Findings of visual neuroscience [2] also show that a combination of several kinds of features is used in primate cerebral cortex for categorization. Features are used in a hierarchical fashion starting from simple features and later using a combination of several features. Our work is motivated by these two observations: firstly it is assumed that there exists a hierarchical relation between object classes and secondly visual categorization is a hierarchical process by using a combination of suitable features.

In general, a hierarchy (e.g. taxonomy) is a result of long time research in the scientific community. The hierarchies, discovered by different researchers, may vary depending on the available data, resources, and method of analysis of data using striking features. Learning of a hierarchy (e.g. taxonomy of biological objects) is a different problem than discovering it. Basically, the learning is a supervised process - it is a process of becoming a specialist in the domain. Such a specialist can be relied on for authentic specification of an unknown instance of an object class (i.e. determining Kingdom, Phylum, Class, Order, ..., Species). In visual object categorization, we learn a given hierarchy by automatically selecting a suitable subset of extractable features and a classification function for each of the superclasses. Once such a hierarchy is learnt the classification method provides a detail specification of an unknown test object (e.g. the given sample is a 'fruit' and an 'apple'). An overview of the method can be found in Fig. 1.

In this work, we have proposed a method of learning a given hierarchy by automatic selection of a set of suitable features for each of the superclasses. The main contribution of this work is to automatically select non uniform features for categorization of diverse class of objects. We consider the superclasses at different levels of hierarchy as independent multi-class classification problems. Then an appropriate classification function (model) is learnt with the selected features. The specification process is carried out in a top-down approach which starts from the root of the hierarchy and follow a single branch of the hierarchy depending on the decision at each internal node until we get a decision about the basic class. This yields a detail specification of the given sample. The proposed method has been tested on different hierarchies and promising results have been obtained.

There are several potential advantages of a hierarchical method [2], [3] such as faster model selection and flexible features choice for improved classification. However, an important objective of hierarchical categorization is to achieve better classification performance on the basic classes. In addition to this we get a detail specification of a

This work was not supported by any organization

given sample with multiple labels. In many applications the ontology of a certain domain may be encoded in terms of object class hierarchy based on their semantic meanings [4]. That is why, hierarchical categorization may be useful for such applications including object classification, semantic image retrieval, image annotation by assigning multiple concepts (i.e. labels) etc.

The outline of the remaining of the paper is as follows. We first review the related work in Section II. We discuss recent advances in hierarchical methods of object classification and learning of such a hierarchy. In Section III we describe our method of hierarchical object categorization. In this section, we also briefly discuss about features and hierarchies used. In section IV, we present the experimental setup, database used, and results. Section V concludes the paper with a discussion of the limitations of current work and our future directions.

II. Related Works

Hierarchical approaches for object categorization have become increasingly popular over the years as it has been regarded as a mean to improve the performance of object classification [3], [5]. Particularly, in pursuit of a general categorization system capable of recognition a vast number of object categories, a need for such a hierarchical structuring has emerged. Hierarchical systems combine existing classes into more complex entities (i.e. superclasses) to achieve more compact object representation enabling fast and robust categorization with better generalization properties [6].

Decision tree [7] based approach is an earlier example to exploit hierarchical feature selection method. Decision tree use heterogeneous and diverse types of features and select them dynamically at different levels. However, this obtained structure does not necessarily follow the semantically meaningful hierarchy (e.g., the subtrees of a decision tree not necessarily correspond to semantically meaningful superclasses.) An extension of this method can be found in [8] where the classification system rather pursuits the hierarchy which is obtained by the semantic information of object class. However, this method uses manually selected features at different levels of hierarchy. In order to improve classification performances, several classification models are also combined together by using them at different levels [5].

The primary concern is how to find the hierarchical relation among the basic classes. Natural hierarchy could be a good choice. Handcrafted hierarchy based on the natural similarity is used for classification in [5]. Hierarchy is constructed through a statistical analysis of elementary similarity in [8]. Some works are also dedicated to discover such a relation automatically. Both supervised and unsupervised learning methods have been investigated. Supervised learning has shown a significant generalization for small sample sets by sharing features between object classes [9]. Recently, unsupervised learning, based on common visual elements also shows comparable performance [4].

III. Hierarchical Categorization

There are two basic steps in object categorization: training and testing. In a regular nonhierarchical object recognition method some train samples for each of the classes are provided. Features are extracted from the training samples and used to learn a classifier (a pre-specified function). For example, *n* SVMs (assuming one-versus-rests approach) are learnt for an *n* class problem. In the testing process, the classifier determines the class of a test object and assigns a label to it. In a simple model, a few specific features (sometimes only one) are used for classification [10], [11]. Recently, automatic feature selection method has emerged [12], where a subset of features from a given set is selected before the training process. However, these selected features remain fixed and are used to learn the classification function. Briefly, in these methods of categorization, all the classes are treated equivalently with the same set of features.

In hierarchical categorization, we assume that a hierarchical relation of the basic object classes is provided in addition to the training samples of each of them. Then, the original classification task is divided into some (hierarchically defined) superclasses. Each of these superclasses is learnt with a suitable subset of features. The two basic steps of hierarchical learning are as follows:

1. Automatic selection of a set of features for each of the superclasses
2. Learning the appropriate classification function for each superclass with the selected features.

Unlike in the regular method, the automatically selected features become non uniform for different superclasses. Once the hierarchy is learnt, a novel test object is categorized in a top-down fashion by assigning multiple labels. Hence we rename the testing process as 'specification' which gives a detail description of the test object. The basic idea of hierarchical categorization is show by the block diagram in Fig. 1.

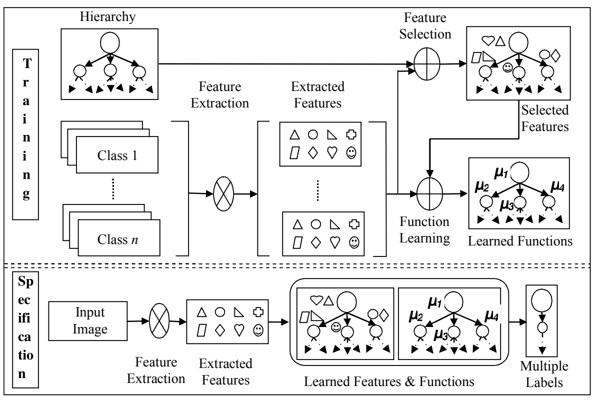


Fig. 1 Block diagram of hierarchical object categorization

We describe the different steps of hierarchical object categorization below. The first step of learning and classification is the extraction of features for respective images. A brief introduction about the image features is given in subsection *A*. A short description of hierarchical representation is given in subsection *B*. The learning and specification methods are described in subsection *C* and *D* respectively. Finally, we discuss the computation cost of the proposed method in subsection *E*.

A. Image Features

For object categorization, we consider locally defined low-level image features. For the recognition of a partially occluded object in cluttered environment, local image features are preferable. These features are generally invariant to different kinds of geometric and photometric transformations as well.

Features are generally extracted from affine covariant interest regions (image patches) by computing different kinds of descriptors of such an image patch. They are represented as a vector of *m* dimensional features space $\Re^m$. An up-to-date review of different methods of local feature detection and description can be found in [1]. All these features are indeed quite successfully used for many applications such as wide baseline matching for stereo pairs [13], content based image and video retrieval (CBIR) from large databases [14], model based object recognition [15], etc. However, it is generally believed that the local-feature-based nearest-neighbor classification method can overcome the intra-class variation of objects to some extent.

Local features are utilized in two different ways in different applications: (i) unquantized features, which comprises two basic steps: feature extraction and feature matching; (ii) quantized bag-of-features and hyperfeatures, which includes the following four steps: feature extraction, feature clustering, frequency histogram construction, and feature matching. Matching approaches depend on distance matrices. For unquantized features, Euclidean distance and Mahalanobish distance are generally preferred. On the other hand Chi-squared (χ^2) distance and Earth Movers Distance (EMD) are popular for histogram comparison [11].

B. Hierarchical Representation

As discussed in section II, three approaches can be identified in the literature to obtain a hierarchy for a given set of object classes: handcrafted, statistical, and learning-based. The final outcome depends on the similarity and dissimilarity among object classes. These similarities and dissimilarities are rather abstract ideas. However, given a measure of similarities and dissimilarities, we can obtain a hierarchy for a give dataset following any of these methods. The hierarchy could vary depending on the designer's perspective and application as well.

In this work, we intuitively group some of the apparently similar classes into a superclass. Two alternative hierarchies are depicted in Fig. 2 for ETH-80 [7] database. As shown in Fig. 2(a); dog, cow, and horse classes are put into a superclass named 'animal' which shares some common shapes. These superclasses always may not have common perceivable shape features rather some other common properties. For example, different types of artefacts may not have any shape similarity; instead they share a common properties of being man-made (e.g. car and cup). Similarly 'small' and 'big' (Fig. 2(b) superclasses have vague shape similarities. These super-classes can be further clustered into the higher-level super-superclasses, and so on. Eventually, we obtain a multilevel hierarchy. For simplicity, we consider only a two-level hierarchy in this work.

A hierarchy can be represented by a general tree where each node can have a different number of children. The root (considered as the first level of decision making) corresponds to the world under consideration. The leaves are the basic classes and each of the internal nodes represents a superclass. Suppose, we have a set of *n* classes $\{C_i \mid i = 1\ldots n\}$, i.e. a tree with *n* leaves. We clusterize the set $\{C_i\}$ into *k* subsets (superclasses) $\{S_j \mid i = 1\ldots k\}$. Thus we form a subtree for each superclass where the root is represented by the superclass, and the basic classes under the superclass become leaves. We may further cluster superclasses into super-superclasses and add another level in the hierarchy and so on. Finally, we add the root which corresponds to the world (database) to form the tree.

C. Automatic Feature Selection

Suppose we have a categorization function (μ) and the input image *X* of an object belonging to class $y \in \{+1, -1\}$. If $\hat{y} = \mu(X)$ then for an ideal classification function we always expect that $|y - \hat{y}| = 0$. As discussed above, the function may have different choices of features. Let us assume we have *l* features $\{f_1, \ldots, f_l\}$ extractable for the function, and an arrangement of features $\alpha(\{f_1, \ldots, f_l\})$ is feasible. We wish to find an α which minimizes categorization error. Now the problem of feature selection is to automatically decide an arrangement α on a given training set such that

$$\arg\min_{\alpha} \sum | y - \hat{y} | \qquad (1)$$

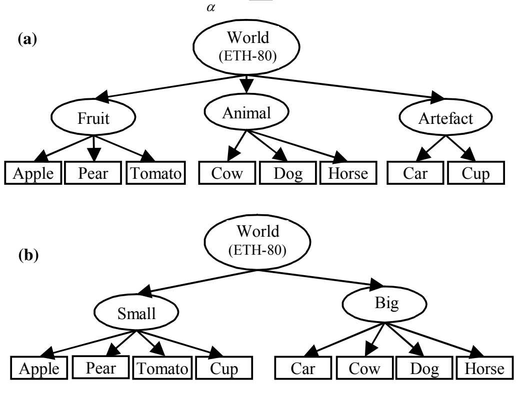


Fig. 2 Two different object class hierarchies for ETH-80 database. (a) HRC-1, (b) HRC-2

The arrangement α could be defined in many ways such as a subset of the features or a weighted contribution of all the features [12], [16].

In this work, we consider the use of SVM classification with multiple kernels as proposed in [12]. Each of the kernels uses a particular feature. The k-th kernel using feature f_k can be written as $K_k(\mathbf{x}, \mathbf{y})=\exp(-\gamma_k f_k(\mathbf{x}, \mathbf{y}))$, where the kernel matrices are strictly positive definite. The basic idea of multiple kernel method is to concatenate the contribution of the features by summing up the individual kernels. The contribution of each of the features is weighted as follows:

$$K = \sum_k d_k K_k \qquad (2)$$

An optimal kernel K is learnt from a given training set by leaning the weight parameters d_k; $k = 1, \ldots, l$. The optimization is carried out in the SVM framework to achieve the best classification on the training set.

D. Learning a Hierarchy

Suppose we are given a hierarchy and a set of training images of each of the n basic classes. The hierarchical structure divides the learning problem into $O(n\text{-}1)$ independent sub-problems. We consider each of the internal nodes (superclasses) including root of the tree as a separate learning problem. Suppose S is such a node having c children. For a child $C_i \in \text{Children}(S)$ we obtain the training set T_i by concatenating the training samples of all basic classes under this branch. Now it becomes a c class problem with T_i training images for each, where $i = 1, \ldots, c$. If we consider the one-versus-rest method, we need to train c SVMs. The learning consists of the following two steps

1. Feature learning: in this step the weights d_k, $k = 1, \ldots, l$ for the features are learned from the training samples. The features having significant contribution are retrained and associated with the node.
2. Function learning: in this step the parameters of c SVMs are learnt and associate them with the node.

The learning process of a hierarchy is summarized in the following Algorithm 1. Here, the input is the hierarchy (H) represented as a tree and the set of training features (F) for all basic classes. The algorithm returns the tree with associated features and functions for each of the internal nodes.

E. Specification

In this step, we obtain a detail specification of a given test object. We start from the root of the hierarchy tree. From the learning process we already know the relevant features for this node and we extract only them for the input object. Then, the associated learned function is used to decide the subclass (the next level super-class in the hierarchy). We repeat the process until we reach a leaf (the basic class). Hence, the method follows a particular branch of height $O(\log_2 n)$ of the tree (i.e. goes through $O(\log_2 n)$ decision processes) yielding a specification with $O(\log_2 n)$ labels.

The specialization method is described by Algorithm 2. Here we supply the hierarchy associated with learned features (H') and classification functions (H'') together with a test object X. The algorithm returns a list of labels denoted by the set S.

Algorithm 1:

```
Hierarchy_Learning (H, F)
    Initilalize(Q)          // initializes an empty queue
    H' = H'' = H;           // initialization
    T = Root(H)             // function Root returns the root of tree H
    EnQ(Q, T)                 // function EnQ enqueues T into the queue
    while Q ≠ Empty
        T = DeQ(Q)     // function DeQ returns top element in the queue
        if Is_Leaf(T) = false  // Is_Leaf determines whether T is a leaf
            for all Ci ∈ children (T)   // for all children of the node T
                EnQ (Q, Ci)
                 // union of all training features of leaves for the subtree
                Fi = Find_all_Training_Features (Ci)
            dk = Learn_weight ({Fi})  // learn weight dk for k = 1, …, l
            α = {□}                    // initialize α (a set of selected features)
            for all k = 1 to l
                if dk > Threshold   // user defined threshold
                α = {α ∪ fk}      // adding significant features in the set
            H' = Associate (α, H', T)   // associate α with the node T
            μ = Learn_Function (α, {Fi}) // learn classifier for the node T
            H'' = Associate (μ, H'', T)  // associate μ with the node T
    return H', H''.
```

Algorithm 2:

```
Specification (H', H'', X)
        T = Root(H')    // function Root returns the root of tree H'
        S = {□}       // initialize the empty set of labels
        while Is_Leaf(T) == false
            // function Is_Leaf determines whether T is a leaf or not.
            α = Get_Ftrs (H', T)   // Get_Ftrs returns learned features at T
            F = Extract_features (α, X) //extracting relevant features
            μ = Get_Func (H'', T)  // Get_Func returns learned function
            T = μ(F)         // decision at current node
            S = S ∪ Label (T)    // append the node label
        S = S ∪ Label (T)         // append the leaf label
        return S.
```

F. Computational Cost

Consider the one-vs-rests method used in a regular nonhierarchical categorization process we need to train n SVMs for a n class problem. Similarly, we need to evaluate n function for testing process. The hierarchical method does not decrease the computational efficiency for the learning process which still remains $O(n)$. However, the real computational cost in the train process is multiplied by a constant depending of the branching factor of the tree. One the other hand, the specification process become $O(\log_2 n)$ instead of $O(n)$ in a nonhierarchical categorization process. This is a desirable property in many cases of categorization problem where training is merely an off-line process but the specification is rather online.

IV. EXPERIMENTAL SETUP

We have tested the hierarchical method and compared the performance with a regular nonhierarchical method of object categorization. We have chosen a multiple kernel learning (MKL) SVM framework with automatic features selection. An implementation of this method is available online (http://www.robots.ox.ac.uk/~vgg/software/MKL). It combines SIFT [17], *self-similarity* [18], and *geometric blur* [10] features with the multiple kernel learning of Varma and Ray [12] to obtain state-of-the art performance. SIFT and self-similarity features are quantized into Bag-of-Words and then spatial histograms are computed. Geometric Blur feature is used without any quantization. Here, seven RBF kernels are combined linearly. For matching, χ^2 distance is used for former two features and correlation-based distance measures [10] are used for the later feature. One-versus-rests SVMs are trained and classification results are obtained by assigning each image to the class that obtains the largest SVM discriminant score.

Our implementation of hierarchical categorization was based on this MKL implementation. Algorithms 1 and 2 were implemented on the top of this implementation with the same set of parameters for the SVMs. For the convenience of comparison, we used the same set of features and kernels. The basic difference was that the features are selected independently for each node of the hierarchy and it was non-uniform. We tested the method on ETH-80 database for the two hierarchies (HRC-1 and HRC-2) in Fig. 2 and compared the results with the nonhierarchical MKL method.

A. Database

The methods were tested on ETH-80 database [7]. In this database, there are eight classes of both biological and artificial objects. In each of the classes there are ten objects. Fig. 3 shows the eight classes with ten objects for each of them. Each object is represented by 41 views with uniform background spaced evenly over the upper viewing hemisphere.

For this database, the suggested test mode is leave-one-object-out crossvalidation. Here all the 79 objects (i.e. 79×41 images) are used as training set and test is carried out with the remaining one unknown object (i.e. 41 images). The results are averaged over all 80 possible test objects.

For more effective evaluation, we carried out experiments with different numbers of training objects. For three different sets of experiments we use two objects, five objects and eight objects of each class. The remaining objects are used as a test set for the respective experiment. For instance, with the eight training objects, for each class we use 8×41 images as the training set and 2×41 images as the test set. In all these experiments we carried out multiple tests for a particular number of training objects. These training objects were selected randomly but kept the same for all the corresponding experiments.

Fig. 3 Ten objects of each of the eight classes in ETH-80 database

B. Results

Fig. 4 shows the recognition rate for both of the methods. This was obtained by averaging the two tests for each experiment with different numbers of training objects. Here we carried out experiments with 2, 5, and 8 training objects. The performance of MKL method is compared with our hierarchical method using two hierarchies (HRC-1 and HRC-2). It can be seen that the hierarchical method has performed somewhat better than the regular MKL method in most of the cases. In these three experiments HRC-1 and HRC-2 are in average superior by 1.0% and 0.8%, respectively, than the nonhierarchical MKL method.

In Fig. 4 the comparison is performed on the basic classes. We have also performed some analysis on the higher levels of hierarchy (only the level just above the leave exists in the given hierarchies). For example, in HRC-1 in Fig. 2(a) there are three superclasses (fruits, animals, artifacts). We compared the recognition error at this level with MKL method. For the nonhierarchical MKL method we obtained equivalent results just by concatenating (bottom-up) the results on basic classes. Here a misclassification within the same superclass is not counted. For example, if an apple is classified as tomato, it is not considered as a misclassification.

Fig. 5 shows the error rate for the second level of hierarchy. Here we need to put the results in separate graphs due to the difference in superclasses in second level. For example, the superclass fruit has three basic classes but the superclass small has four. Thus, to make the result comparable with MKL method, we concatenate the results of the respective three and four classes. It can be seen that hierarchical methods results less error in the higher level of hierarchy. It means that if we wish to retrieve objects of a superclass (e.g. fruits) we will achieve better accuracy. In these three experiments HRC-1 and HRC-2 have average error rates lower by 2.7% and 2.3%, respectively, compared to a nonhierarchical MKL method.

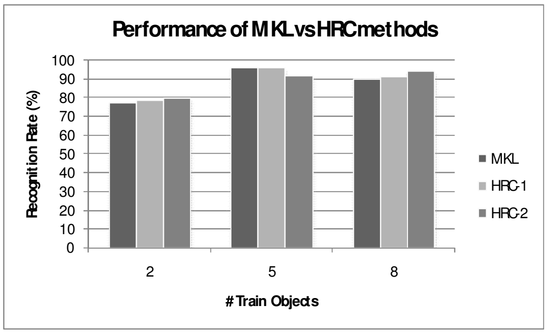
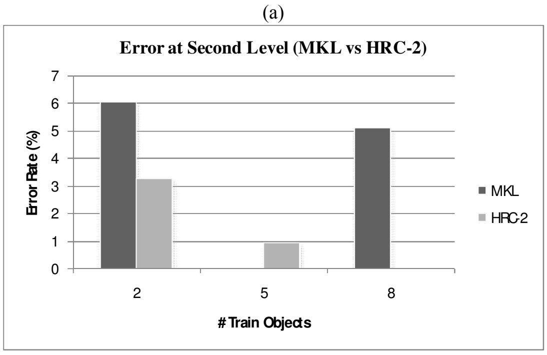


Fig. 4 Comparison of recognition rates between MKL and hierarchical methods

(a)



(b)

Fig. 5 Comparison of error rate between MKL and hierarchical method at second level

V. CONCLUSION

This paper describes a method of automatic feature selection for hierarchical object recognition. Automatic feature selection and hierarchical object categorization techniques have evolved recently. We are motivated by biological vision systems to put them together to improve recognition performance. In this work, we have tested our proposed method on two hierarchies, given for an ideal database. The hierarchies are based on natural similarities and properties of the basic object classes. We wanted to investigate feasibility of the proposed method. In fact, the experimental results are found consistent with our expectation. With the same parameters of a SVM based classification framework, we achieved better categorization performance without any increase in computational costs. This was true for both of the hierarchies. More importantly, we obtained less error in categorization on the superclass level. All these implicate the potentiality of this method especially in object categorization, semantic retrieval, automatic annotation, and other areas.

At the primary stage, we have tested our method on a small database. Here we assumed that the objects in all basic classes and superclasses are unimodal. This is an unrealistic assumption indeed. So trying with multimodal classification method at different levels of hierarchy should further improve the categorization performance. Furthermore, we have used the same classification model at different levels of hierarchy. A method which is capable to automatically select different models of categorization could further improve the results.

ACKNOWLEDGMENT

The research presented in this paper is a part of A*STAR Science \& Engineering Research Council grant 072 134 0052. The financial support of SERC is gratefully acknowledged.

REFERENCES

[1] J. Li and N. M. Allinson, "A comprehensive review of current local features for computer vision," Neurocomputing, vol. 71, pp. 1771-1787, 2008.

[2] E. Rolls and G. Deco, Computational Neuroscience of Vision: Oxford University Press, 2002.

[3] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: learning a hierarchy of parts," in IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, 2007, pp. 1-8.

[4] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros, "Unsupervised discovery of visual object class hierarchies," in IEEE Conference on Computer Vision and Pattern Recognition 2008, pp. 1-8.

[5] A. Zweig and D. Weinshall, "Exploiting object hierarchy: combining models from different category levels," in IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1-8.

[6] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in The 10th IEEE International Conference on Computer Vision 2005, pp. 1331 - 1338.

[7] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in IEEE Conference on Computer Vision and Pattern Recognition Madison, USA, 2003, pp. 409-415.

[8] I. Autio, "Using natural class hierarchies in multi-class visual classification," Pattern Recognition, vol. 39, pp. 1290-1299, 2006.

[9] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Describing visual scenes using transformed objects and parts " International Journal of Computer Vision, vol. 77, pp. 291-330, 2008.

[10] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2126- 2136.

[11] J. Zhang and M. Marszalek, "Local features and kernels for classification of texture and object categories: a comprehensive study," International Journal of Computer Vision, vol. 73, pp. 213 - 238, 2006.

[12] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in IEEE International Conference on Computer Vision, 2007.

[13] T. Tuytelaars and L. V. Gool, "Matching widely separated views based on affine invariant regions," International Journal of Computer Vision, vol. 1, pp. 61-85, 2004.
[14] J. Sivic and A. Zisserman, "Video data mining using configurations of viewpoint invariant regions," in IEEE Conference on Computer Vision and Pattern Recognition, 2004, pp. 488-495.
[15] V. Ferrari, T. Tuytelaars, and L. V. Gool, "Simultaneous object recognition and segmentation by image exploration," in Eight European Conf. Computer Vision, 2004, pp. 40-45.
[16] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1 - 8.
[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, pp. 91-110, 2004.
[18] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in IEEE Conference on Computer Vision and Pattern Recognition, 2007.