Semantic Hierarchies for Visual Object Recognition

Marcin Marszalek, Cordelia Schmid

Semantic Hierarchies for Visual Object Recognition

Marcin Marszałek Cordelia Schmid
INRIA, LEAR - LJK
665 av de l'Europe, 38330 Montbonnot, France
Marcin.Marszalek@inrialpes.fr Cordelia.Schmid@inrialpes.fr

Abstract

In this paper we propose to use lexical semantic networks to extend the state-of-the-art object recognition techniques. We use the semantics of image labels to integrate prior knowledge about inter-class relationships into the visual appearance learning. We show how to build and train a semantic hierarchy of discriminative classifiers and how to use it to perform object detection. We evaluate how our approach influences the classification accuracy and speed on the PASCAL *VOC challenge 2006 dataset, a set of challenging real-world images. We also demonstrate additional features that become available to object recognition due to the extension with semantic inference tools—we can classify high-level categories, such as animals, and we can train part detectors, for example a window detector, by pure inference in the semantic network.*

1. Introduction

The recognition of object categories in images is one of the most challenging problems in computer vision, especially when the number of categories is large. Humans are able to recognize thousands of object types, whereas most of the existing object recognition systems are trained to recognize only a few. In this paper we address two important limitations for constructing vision systems which deal with a large number of categories: a) inter-class similarities and relationships need to be modeled; b) the complexity in the number of object categories has to be reduced.

Both points are addressed in the following by incorporating prior knowledge about object identity into the visual recognition system. Humans use this knowledge when learning the visual appearance of the objects [9]. For instance, when one encounters a new car model, it is not sensible to learn all the appearance details. It is enough to remember that it looks like a car as well as the discriminative details. This can help to learn the visual appearance of new object types and speed up the recognition process—both advantages are very desirable in object recognition. Moreover, by generalizing over object instances, humans can say something meaningful about the appearance of each of the terms forming a hierarchy like: *Maybach* → *car* → *motor vehicle* → *vehicle* → *artifact*[1] → *physical object*. This allows to learn new concepts by semantic inference and to give richer answers due to possible reasoning—it is again useful to bring those features to object recognition. The above mentioned facts inspired us to collect the semantic knowledge starting from the semantics encoded in the class labels and mimic the described behavior in machine vision.

Related work. Existing object recognition techniques rarely consider inter-class relationships, i.e., they treat the classes as completely separate and independent both visually and semantically. For example, the method that consecutively won the detection task of the two recent PASCAL Visual Object Classes challenges [4, 5] performs multi-class detection with a set of binary SVM classifiers in the one-against-rest setting [18]. With the growing number of categories this is not only ineffective, but can also lead to training a "cars vs Maybachs, vehicles, all-the-possible-objects" classifier, as it ignores the semantic relationships between classes which exist in the real world. Moreover, *cars* and *buses* are for example more related to each other than *dogs* and *bicycles*, which is also missed.

Knowledge can be modeled by ontologies. For example, lexical semantic networks are used to model human psycholinguistic knowledge. One of the most popular semantic networks for English language is WordNet [6]. It groups words into sets of synonyms and records different semantic relations between them. This allows to infer, for example, that a *car* is a *wheeled vehicle* and that a *motorcycle* is also a *wheeled vehicle*, thus both should incorporate a *wheel*. Querying the semantic network of WordNet, one can determine semantic relationships between class labels that are assigned to the observed visual object instances during visual object recognition.

Linguistic relations between annotations have been successfully exploited in image retrieval [1, 17]. While we share the idea of using WordNet to find semantic relationships between class labels, we go beyond completing the annotations or extending the queries. As we show in the

[1] By the *artifact* we mean a man-made object.

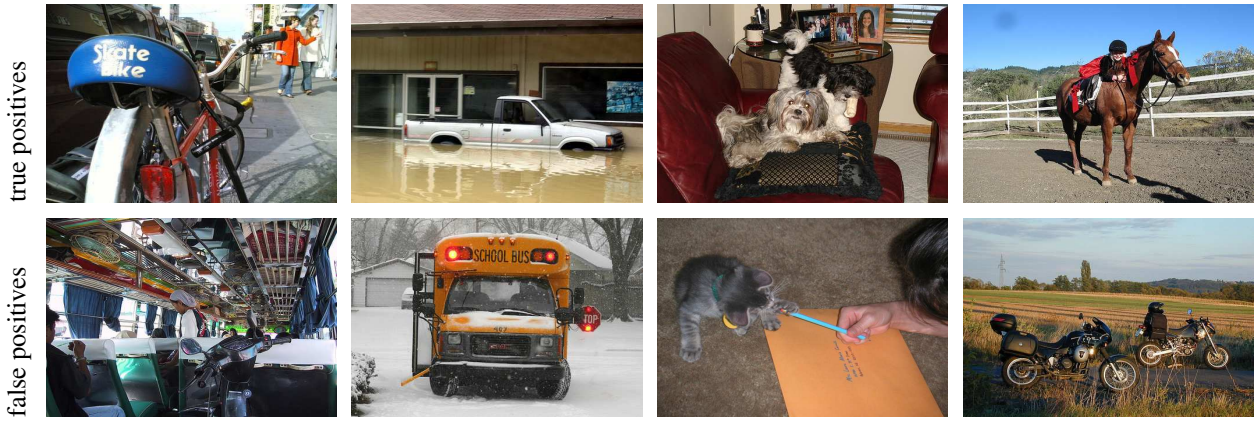


Figure 1. Sample PASCAL VOC'06 images classified with our semantic hierarchic classifier.

experimental section, incorporating the semantics into the knowledge representation leads to better recognition accuracy than relying only on straightforward reasoning, as it also allows to discover additional visual cues that would be missed otherwise. Semantic hierarchies have proved to be useful for automatic image annotation [14]. We use them to combine discriminative classifiers and thus choose a different strategy for exploiting their structure. We will demonstrate that this introduces some additional features for object detection, in particular it allows to give sensible answers in situations of uncertainty and to learn new classifiers using inference. We also go beyond the ISA relationships taking advantage of PARTOF and MEMBEROF relationships.

Overview of our method. The problem of object recognition is often given in the form of a classification task, an assignment problem in which a semantic term encoding the object identity (a label) has to be assigned to an observed visual object instance. The classification problem can be extended to a detection problem, where instead of questions like "Is it a car?" we answer questions like "Is there a car?". The detection task usually assumes not only background clutter, but also permits co-occurrence of multiple object instances, even representing different object classes. Thus, unlike the classification task, it is often multi-label. Note that we further distinguish the detection task from the localization task, where additionally the locations of the objects have to be given. In this paper we focus on the detection task, but our research can be directly applied to image classification and also incorporated into object localization methods.

The combination of bag-of-features image representation [16] with Support Vector Machines [13] (SVMs) resulted in successful object recognition methods [5, 18]. We take the state-of-the-art image classification method of Zhang et al. [18] to implement the underlying binary classifier for our method. To create the semantic hierarchic classifier for object detection, we query the WordNet with the class labels and extract the knowledge in the form of a semantic hierarchy. This hierarchy is used for reasoning and to organize and train the binary SVM detectors. The trained hierarchic classifier can be used to efficiently recognize a large number of object categories. This is explained in detail in section 2. Section 3 presents the experimental results on the natural-scene PASCAL VOC'06 dataset [4] (see fig. 1 for sample results obtained with our method). In subsection 3.1 we compare the performance of our classifier to the state-of-the-art, whereas in subsection 3.2 we discuss the additional features of our classifier. We conclude the paper in section 4.

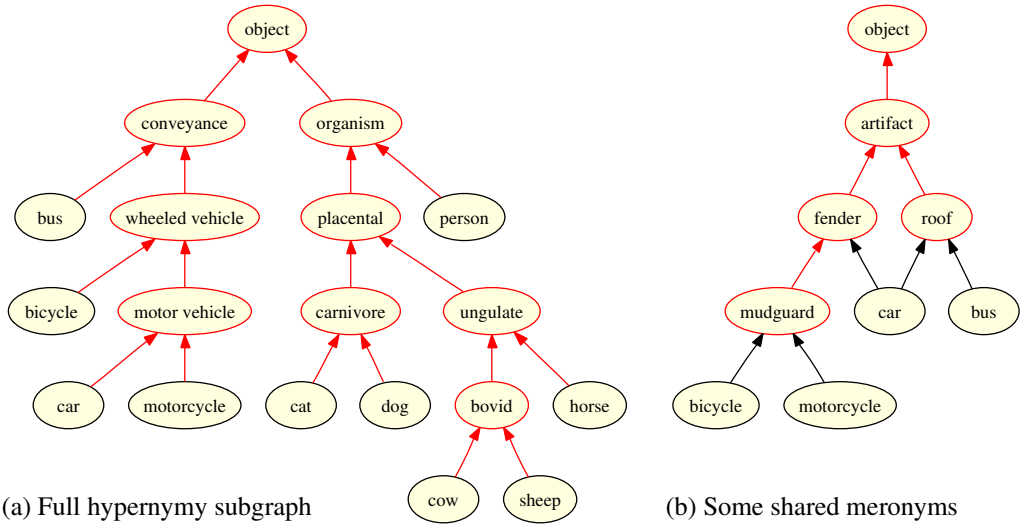
2. The semantic hierarchic classifier

We first describe the two key elements of our system—the underlying binary detector (subsection 2.1) and the extracted semantic graph (subsection 2.2). Then, in subsection 2.3, we explain how to merge those elements to obtain the semantic hierarchic classifier.

2.1. Detecting the presence of a given class

In the following we describe the object detection framework of Zhang et al. [18]. Given an image, we use two complementary scale-invariant local region detectors to extract salient image structures: the *Harris-Laplace* detector [12] responds to corner-like regions and the *Laplacian* detector [10] extracts blob-like regions. To compute appearance-based features of the patches extracted by the detectors, we use a combination of the *SIFT* [11] descriptor and the *hue* color description [15]. The SIFT descriptor is based on a grayscale gradient orientation histogram of dimension 128 and the color description is a hue histogram of dimension 36, i.e., the combined descriptor is of dimension 164.

We first build a visual vocabulary by clustering the feature vectors from the training images. Our experiments have shown that vocabulary construction has little impact



(a) Full hypernymy subgraph (b) Some shared meronyms

Figure 2. WordNet 2.1 subgraphs for the VOC'06 labels. Intermediate nodes were removed for clarity. Note the obvious *bus* synset (concept) misplacement.

on the final results. We, therefore, randomly subsample a set of 50k features and cluster them using k-means with $k = 1000$. This results in a vocabulary consisting of 1000 visual words. Given a vocabulary, we can represent each image in the dataset as a histogram of visual words [16]. Each feature of image i is matched with the closest word in the vocabulary based on the Euclidean distance. Each histogram entry $h_{ij} \in H_i$ is then the proportion of all descriptors in image i matched with vocabulary word j with respect to the total number of descriptors computed for the image.

We use SVMs [13] with an extended Gaussian kernel [2] $K(H_i, H_j) = e^{-\frac{1}{A}D(H_i,H_j)}$ for classification, where H_i and H_j are image histograms and $D(H_i, H_j) = \frac{1}{2}\sum_{n=1}^{k}(h_{in} - h_{jn})^2/(h_{in} + h_{jn})$ is the χ^2 distance. The resulting χ^2 kernel is a Mercer kernel [7]. The parameter A is the mean value of the distances between all training images [18]. We combine different detector/descriptor channels by summing the corresponding distances, such that $D = \sum_n D_n$ where D_n is the χ^2 distance for channel n.

2.2. Extracting the semantic graph from WordNet

WordNet 2.1 [6] contains over 80000 noun synonym sets called *synsets*. A synset is a set of words that are interchangeable in some context without changing the truth value of the preposition in which they are embedded. If a given word has more than one meaning, it may belong to more than one synset, but for each sense exactly one synset is defined. Thus, synsets model concepts and are represented with nodes in the semantic graph. Between synsets semantic relationships are defined. Between nouns, antonymy (opposition in meaning), hypernymy/hyponymy (superterm/subterm) and holonymy/meronymy (is a part of/contains) relationships are possible. A synset can also create a domain (a topical class), to which other synsets are linked. Semantic relations are represented with directed edges (links) in the semantic graph. For the detection task, strong reasoning is possible using hypernymy ("If there is a car then there is a vehicle") and meronymy ("If there is a car then then there is a wheel"). For classification task antonymy could be used ("If it is a man then it is not a woman"), but this cannot be generalized to the detection task (as there can be both a man and a woman in one image). Domains cannot be used for strong reasoning (a presence of a passenger does not assure the presence of a bus, nor does the inverse hold)—they could, however, be used for weak reasoning.

As we focus on detection and need strong reasoning for training the hierarchic classifier, we first extract from the WordNet the synsets that correspond to the class labels and then follow the hypernymy and meronymy links to obtain the subgraph. Some researchers consider only hypernymy/hyponymy (ISA relationship) for reasoning [14, 17], but we find that incorporating holonymy/meronymy (PARTOF and MEMBEROF relationships) permits much richer reasoning. When extracting the VOC'06 labels and following only hypernymy links, the resulting subgraph contains 42 nodes. If we also follow meronymy links, the graph contains 1452 nodes.

Fig. 2 presents the subgraph of WordNet which corresponds to VOC'06 labels. The complete graph is shown in the case of the hypernymy relationships (left), except the intermediate nodes which are removed for clarity. We can see that the WordNet query results in a reasonable semantic hierarchy that mostly reflects visual similarities. Interestingly,

the *person* is not placed under the *placental*. This is due to the fact that WordNet reflects psycholinguistic and not strict scientific knowledge. Following the meronymy links enriches the graph. Some meronyms shared between labels are shown in fig. 2 (right). Unfortunately, there are also some errors and inconsistencies. The unexpected placement of the *bus* synset is due to the missing hypernymy link to *motor vehicle*. A meronymy link from *bus* to *fender* is also missing. Note that for clarity of presentation we have used only one relation for each of the graphs shown. When we follow more types of links, the resulting graph interleaves all relations considered. Furthermore, the resulting hypernymy graph is a binary tree, but in general more subnodes and even many supernodes are possible and supported by our method (we assume a DAG, which holds for WordNet).

Following meronymy links without any limitations unfortunately permits reasoning which is incorrect from the point of view of visual appearance. For instance, a car contains fuel in its tank, which is an organic material. This, however, does not imply similarity to living organism like a cat. To prevent reasoning through unobservable entities like substances, we implement a pruning process. First, a *base node* is found—a "minimal" node from which all the synsets corresponding to the queried labels can be reached through hyponymy links. Then, after performing the full WordNet query, the nodes that cannot be reached from the base node through the hyponymy links are removed from the graph. In case of our experiments, pruning reduced the number of nodes from 1452 to 563 and assured reasonable inference from the viewpoint of visual appearance.

It is worth noting, that we are guaranteed to find a base node, as any noun ISA *entity*. In theory more than one node could serve as the base node, but in practice usually only the *object*[2] synset satisfies the criterion. Another interesting feature of pruning through the base node is that it adapts the whole graph to the domain of the queried labels. If the labels would refer to various animals, the *animal* synset would be found as the base node and any non-animal parts and members would be automatically rejected when reached.

2.3. Constructing the semantic hierarchic classifier

In order to explain the construction of the semantic hierarchic classifier, we first discuss a model framework in which a discriminative SVM classifier (cf. subsection 2.1) is associated with each edge of the obtained semantic graph (cf. subsection 2.2).

Let us start with looking for images (exemplars) supporting a given concept. Trivially, the exemplars that represent the concept itself will support it. Due to the strong reasoning, however, each node of the semantic graph is additionally supported by the union of the exemplars supporting the nodes that point to it through hypernymy or meronymy links, i.e.,

$$\mathrm{supp}(A) = \bigcup_{B_i \to A} \mathrm{supp}(B_i) \cup \mathrm{lbl}(A) \qquad (1)$$

where $\mathrm{supp}(A)$ is a set of exemplars supporting the A concept, $B_i \to A$ is true when B_i links to A through hypernymy or meronymy and $\mathrm{lbl}(A)$ is a set of exemplars labeled with the A concept. For instance (cf. fig. 2), whenever we observe a car or a motorcycle, we observe at the same time a wheeled vehicle, a motor vehicle, a means of transportation, etc. Also, whenever we observe a bicycle or a motorcycle, we observe a mudguard, a wheel, etc.

We train a given $B_i|A$ classifier associated with the $B_i \to A$ hypernymy or meronymy edge by training a binary SVM classifier with

$$P = \mathrm{supp}(B_i) \qquad N = \mathrm{supp}(A) - \mathrm{supp}(B_i) \qquad (2)$$

where P is the set of positive training exemplars and N is the set of negative ones. Given a test sample and knowing that it represents the A concept, we can then consider descending through hyponymy and holonymy links to B_i. We do so, when the detector associated with the $B_i \to A$ link returns a positive answer. For instance, if we know that a test image satisfies the *organism* concept, we can check whether it satisfies the *person* concept by running the *person*|*organism* classifier distinguishing between people and other organisms like animals.

The *base node* is by definition supported by all the exemplars in the dataset. Making an assumption that the training set reflects the test data, we conclude that on any test image the base concept is present. Thus, we can start our classification at the base node and then descend the semantic hierarchy. For instance, we know that any image of the VOC'06 dataset contains an *object*. Then for the *object* node we can have an *artifact* detector and an *organism* detector. For a detected *organism* we can launch *person* and *animal* detectors. After *artifact* detection we can look for windows and means of transportation, and so on.

Please note, that if $\mathrm{supp}(A) = \mathrm{supp}(B_i)$ then $N = \varnothing$. This is often the case, as there are many intermediate nodes without their own labels[3] which are linked from nodes with exactly the same support (often there is only one linking node). Such situation results in a trivial detector that would for instance claim (cf. fig. 2) that every *animal* is a *placental* because it has never seen an animal that would not be a placental. Still, if the training data represents the test data, this is a good conclusion. To avoid passing through the trivial

[2] The *object* synset is defined as a visible entity.

[3] Actually, in case of most object classification datasets available nowadays, only leaf nodes are labeled. Our theory, however, fully supports situations where classes are overlapping, labeling is incomplete, etc.

detectors, we reconstruct the originally obtained semantic graph in a manner similar to subset construction [8]. We define a *conset* as a set of nodes (concepts) with the same support, thus the support of a conset is equal to the support of any of its elements. Given a conset, we can extend it through trivial (leading to nodes with the same support) hyponymy and holonymy links. A maximally extended conset may lead to several nodes with different supports. We group the connected nodes with the same support into new consets and train an SVM classifier for each link to a new conset according to eq. (2). By first extending the conset consisting of a *base node* and then recursively extending the connected consets, we create a hierarchic classifier.

Given a test sample, we start at the *base conset* (extended from the base node) and descend to the linked consets when the classifier returns a positive answer. When reaching a conset, we can label a test image with all the concepts (synsets) belonging to the conset. Note, however, that usually the *intermediate* concepts with only trivial links in the original semantic graph are probably less interesting from the point of view of the user than the *boundary* concepts that link to the concepts with different supports. The concepts that point through hyponymy or holonymy links to the concepts belonging to different consets give the most precise explanation of the sample from the point of view of a given conset. In parallel, the concepts at the boundary through hypernymy or meronymy links give the most abstract (still, however, limited to a given conset) explanation of the sample. Note that given a hierarchy, the boundary concepts determine the intermediate ones.

It is difficult to give a bound on the complexity of our semantic hierarchic classifier. The total number of binary classifiers evaluated depends not only on the structure of the hierarchy which may vary significantly from one set of labels to another, but it also depends on the difficulty of the test images which influence the number of paths considered simultaneously. We can estimate the number of binary classifiers evaluated for a test sample with:

$$T(n) = \frac{c}{a}\, T\Big(\frac{n}{b}\Big) + c \tag{3}$$

where n is the number of classes, c is the number of binary classifiers evaluated at a node, a is the subproblem selection factor (c/a defines the number of subproblems that have to be solved) and b is the problem reduction factor (n/b defines the size of the subproblem). b and c depend on the semantic hierarchy structure, a depends on the complexity of the test image. Thus, b and c parameters vary from node to node and the a parameter depends on the test sample, but we can average them for a given dataset. In the case of the VOC'06 dataset, there were on average $c = 2.85$ subproblems considered (binary classifiers evaluated) per node. Among those, one of every $a = 1.94$ subproblems were descended while the size of the problem was reduced $b = 1.82$ times. According to the master theorem [3], this allows us to estimate the complexity of our classifier (for similar datasets) as:

$$T(n) \in \Theta\Big(n^{\log_b(c/a)}\Big) \approx \Theta\big(n^{0.64}\big) \tag{4}$$

when $\log_b(c/a) > 0 \Leftrightarrow c > a$ which is true. This is significantly better than $\Theta(n)$ required in a one-against-rest setup with n classifiers.

3. Experimental results

We evaluate our semantic hierarchic classifier on the PASCAL VOC'06 dataset [4]. The dataset contains 1277 training images and 1341 validation images which we used for testing. Each image contains one or more objects and each object is annotated as belonging to one of the 10 predefined classes (*bicycle*, *bus*, *car*, *cat*, *cow*, *dog*, *horse*, *motorcycle*, *person*, *sheep*). Sample images are shown in fig. 1. The detection task requires to automatically determine objects classes which are present in a test image. We train our system providing a list of object classes which are present in each image without indicating their locations.

The results of the different methods are evaluated with the Equal Error Rates (EERs)[4] of the Receiver Operating Characteristic (ROC) curves on a per-class basis [5]. To compute the ROC curve our classifier has to return confidence values. In the case of the binary SVM classifier, we use the absolute value of the decision function. For hierarchic classifiers we combine the decision functions of the underlying binary classifiers, i.e., for each concept c we define a decision function $h_c(x)$:

$$h_c(x) = \max_{v \ni c} \max_{P \in \mathcal{P}(s,v)} \min_{e \in P} g_e(x) \tag{5}$$

where v is a *conset* (classifier node) containing the concept c, $\mathcal{P}(s, v)$ is the set of all possible paths from the *base conset* (starting node) s to conset v, P is an element of this set, e is an edge on the path P and $g_e(x)$ is the decision function of the classifier associated with the edge e. In other words, for a given class c and sample x, the maximal decision value over all possible classification paths is returned, whereas for a given path the minimal decision value over its edges is chosen.

Table 1 compares the performance of our semantic hierarchic (SH) classifier with the performance of a standard one-against-rest (OAR) classifier and a classifier based on an automatically constructed visual hierarchy (AVH). AVH is a binary tree obtained through iterative merging of the classes with the smallest average χ^2 distance between their exemplars, i.e., presumably the most visually

[4]Precisely, the point where the recall is equal to the precision is called *break even point*. For consistency with the literature we denote it as EER.

		baseline		our SH		
		OAR	AVH	SSH	ESH	gain
A	bicycle	79.3%	80.0%	81.4%	**82.8%**	**3.4%**
	cat	**82.5%**	**82.5%**	80.4%	80.4%	-2.1%
	sheep	82.6%	81.8%	**84.1%**	**84.1%**	**1.5%**
	average	82.19%	82.02%	**82.52%**	**82.53%**	**0.34%**
B	conveyance	*89.8%*	*88.4%*	**90.4%**	**90.4%**	**0.6%**
	organism	*76.2%*	*82.1%*	**87.7%**	**87.7%**	**11.5%**
C	window	*62.5%*	*62.5%*	-	**65.8%**	**3.3%**

Table 1. Comparison of the EERs achieved by the different classifiers. Sections A and B evaluate the performance on the PASCAL VOC challenge 2006. Section C tests the generalization ability of the classifiers using an external set of images.

similar ones. In the case of both baseline classifiers, OAR and AVH, the same underlying binary classifiers were used as in our SH classifiers. The proposed semantic hierarchic classifier was evaluated in a simple form (SSH), where only hypernymy/hyponymy relationships were used, as well as in an extended form (ESH) that also includes meronymy/holonymy.

3.1. Comparison with existing solutions

Section A of table 1 shows the results for the VOC'06 task of detecting ten object categories. The average over all classes and individual results for classes for which at least 1.5% difference between OAR and ESH classifiers was observed are given. Our approach leads on average to a slightly better performance than the methods that do not use the semantics. This is an encouraging result, as this means improved classifier complexity and additional features (cf. section 3.2) without loss of accuracy. Note that the visual hierarchy usually shows worse performance than the semantic one. This means that apparent visual similarities between images may not always generalize well to object classes and using external semantic knowledge can help to better discover the visual properties of object classes. Fig. 1 presents some true positives and false positives. We can see that our classifier performs well even for images with unusual object views and that it makes mistakes in situations where a lot of context information is necessary to return the correct answer.

It is worth adding, that our semantic hierarchic classifier performs comparably to the state-of-the-art. In the VOC'06 challenge [4], the average EER of the winning method QMUL_LSPCH was 86.4%. Our method achieves an average of 82.5% with half of the training images (we have used the validation set for testing) and including the images marked as "difficult", skipped for the evaluation of the submissions to the VOC'06 challenge.

Furthermore, we anticipate that the gain should further increase for a large number of categories, as the inter-class similarities and overlaps will cause more and more confusion of classifiers devoid of semantics.

3.2. Additional features of our SH classifier

Adding semantic reasoning to a visual object recognition system opens several new possibilities. Firstly, it allows to complete the labels in the training set and permits reasoning to enrich the answers. For instance, the images marked only as *Maybach* could be used for training a *car* detector and after detecting a *car* in a test image the image can also be labeled as *motor vehicle*. Our semantic hierarchic classifier performs both types of reasoning implicitly and thus fully supports incomplete labeling and overlapping classes (like *cars* and *vehicles*). Secondly, the semantic hierarchic structure of our classifier provides sensible answers in uncertain situations. For instance, even when the classifier does not know whether there is a cat or a horse in the image, it may still be certain that there is a living organism and thus provide useful information.

Section B of table 1 compares the results of detecting the high-level concepts *organisms* and *conveyance* for training performed with the original labels. OAR and AVH methods are not capable of reasoning, so a simple form of it ("If there is a cat then there is an organism", etc.) was artificially added after the object detection phase. Our hierarchic classifier directly labels the test images with those concepts and the achieved results are significantly better. The fact that AVH outperforms OAR in the case of the *organisms* suggests that the classifiers get easily confused by different creature types. The observed 11.5% gain when comparing ESH to OAR shows, however, that our classifier goes beyond the aforementioned reasoning and is able to successfully detect a living creature without being explicitly aware of any of the creatures known. We conclude that our classifier is able not only to perform reasoning (none of the training images were marked with the tested concepts), but also to better organize the collected knowledge about the visual appearance of the objects.

In the previous experiment all the living creatures in the test set were corresponding to the creatures from the training set. To test the true generalization ability of our classifier we have collected 120 *vehicle window* images by querying Google Image Search with "vehicle window", "wind-

(a) true positives (b) false positives

Figure 3. Sample *vehicle window* and *machine* images classified by our semantic hierarchic classifier as containing a *window*. Training was performed on VOC'06 images with the original annotations.

screen" and "windshield" queries and manually validating the returned images. For the negative set we have collected 120 images retrieved with the "machine" query. Section C of table 1 shows the results for these test images; training was performed on the VOC'06 images with the original labels. For the OAR and AVH methods post-classification reasoning was performed ("if there is a car or bus than there is a window"). The SSH classifier could not perform the task as it was trained without meronymy/holonymy relationships. Our ESH classifier shows significantly better performance than the methods only extended with reasoning. This confirms that the classifier was able to generalize over the windows of cars and buses. Fig. 3 illustrates examples for detecting individual windscreens and windows of different vehicles. Even some false positives on window-like structures were observable, see fig. 3. We conclude, that our classifier is able to learn the generalized visual appearance of unseen object classes through inference.

4. Summary

In this paper we have proposed a semantic hierarchic classifier that uses the semantics of image labels to extract knowledge about the inter-class relationships and that integrates it into the visual appearance learning procedure. This allows to reduce the classifier complexity in the number of classes and, as was shown in the experimental section, helps to learn the visual similarities. We have also demonstrated additional features of our classifier like returning valuable information in situation of uncertainty and learning new classifiers through inference. The classifier's ability to support overlapping classes and provide a complexity that is sublinear in the number of classes makes it suitable for object recognition tasks that require recognizing a large number of categories. Future research could focus on adding support for weak reasoning.

Acknowledgments

M. Marszałek is supported by a grant from the European Community under the Marie-Curie project VISITOR. This work was supported by the European funded research project CLASS and the EU network PASCAL.

References

[1] Y. Aslandogan, C. Thier, C. Yu, J. Zou, and N. Rishe. Using semantic contents and wordnet in image retrieval. In *SIGIR*, 1997.

[2] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *NN*, 1999.

[3] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

[4] M. Everingham, L. V. Gool, C. Williams, and A. Zisserman. The 2006 PASCAL visual object classes challenge. In *The PASCAL Visual Object Classes Challenge Workshop*, 2006.

[5] M. Everingham, A. Zisserman, C. Williams, L. V. Gool, et al. The 2005 PASCAL visual object classes challenge. In *First PASCAL Challenge Workshop*. 2005.

[6] C. Fellbaum, editor. *Wordnet: An Electronic Lexical Database*. Bradford Books, 1998.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *PAMI*, 2004.

[8] J. Hopcroft and J. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.

[9] P. Jolicoeur, M. Gluck, and S. Kosslyn. Pictures and names: making the connection. *Cognitive Psychology*, 1984.

[10] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 1998.

[11] D. Lowe. Distinctive image features form scale-invariant keypoints. *IJCV*, 2004.

[12] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 2004.

[13] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[14] M. Srikanth, J. Varner, M. Bowden, and D. Moldovan. Exploiting ontologies for automatic image annotation. In *SIGIR*, 2005.

[15] J. van de Weijer and C. Schmid. Coloring local feature extraction. In *ECCV*, 2006.

[16] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *IWLAVS*, 2004.

[17] C. Yang, M. Dong, and F. Fotouhi. Learning the semantics in image retrieval - a natural language processing approach. In *CVPRW*, 2004.

[18] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 2007.