All Vehicles are Cars: Subclass Preferences in Container Concepts

Daan T. J. Vreeswijk Koen E. A. van de Sande
Cees G. M. Snoek Arnold W. M. Smeulders

Intelligent Systems Lab Amsterdam, Informatics Institute, University of Amsterdam
Science Park 904, 1098XH Amsterdam, The Netherlands
{d.t.j.vreeswijk,k.sande,cgmsnoek,a.w.m.smeulders}@uva.nl

ABSTRACT

This paper investigates the natural bias humans display when labeling images with a container label like *vehicle* or *carnivore*. Using three container concepts as subtree root nodes, and all available concepts between these roots and the images from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, we analyze the differences between the images labeled at these varying levels of abstraction and the union of their constituting leaf nodes. We find that for many container concepts, a strong preference for one or a few different constituting leaf nodes occurs. These results indicate that care is needed when using hierarchical knowledge in image classification: if the aim is to classify *vehicles* the way humans do, then *cars* and *buses* may be the only correct results.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Thesauruses; I.4.8 [**Scene Analysis**]: Object Recognition

General Terms

Algorithms, Experimentation

Keywords

Large scale image recognition, Hierarchical image recognition, Classifier combination

1. INTRODUCTION

The number of object categories in the world is hard to count, but it may be in the same order of magnitude as the number of words in language. For daily speech, a well-educated native speaker of English has an active vocabulary of about 17,000 words [4], but surely that is not enough to identify all concepts. The Oxford English Dictionary contains some 600,000 words[1], but since some concepts will be indicated with a combination of a noun and an adjective, like *self-propelled vehicle* and *lame duck*, and many words have more than one meaning depending on the context, the number of concepts must be much larger than the number of words in language. And, beyond words, visual concepts make distinctions where words fail. A "horse on a beach" is likely to depict something different from a "horse with a plough". A "car with a straw hat" is likely to be rather different from a "car with sirens on".

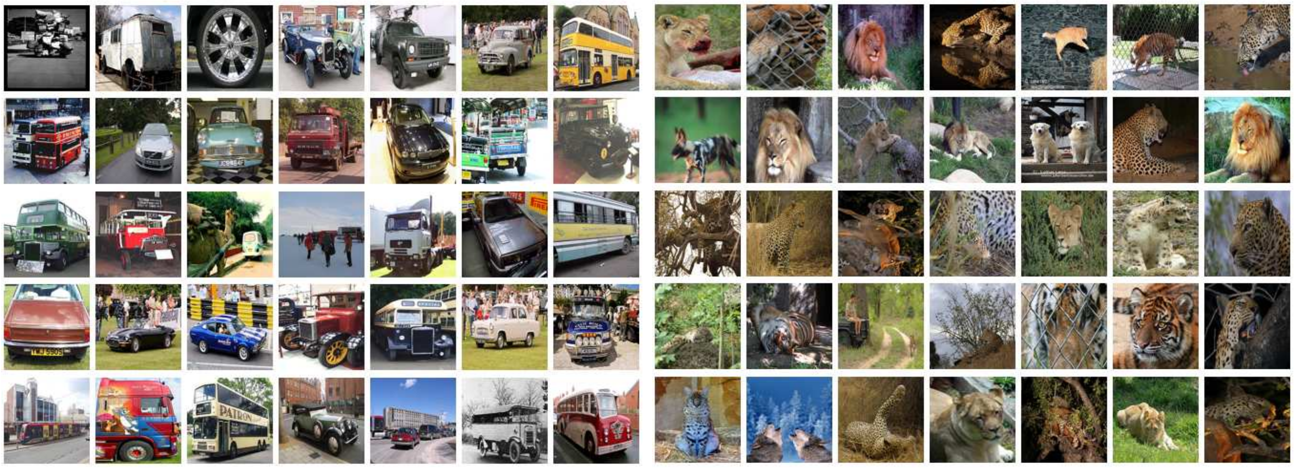
In fact, the space of concepts is named by the hierarchical order which comes with language, refining the denomination of the objects in the world around us further and further. This hierarchy, and its potentially endless refinement, makes the number of concepts countless and in the end less relevant. More relevant in the understanding of the denomination is the hierarchical structure of concepts. Language brings more than one hierarchy. One common hierarchy in language is the order of things by function: *cars* and *bicycles* are both *vehicles*. Other hierarchies are by material (*iron* covers *steel* and *cast iron*), by biological classification (a *cow* and a *monkey* are both *mammals*) or by human definition (e.g. *speed skating* and *football* are both *sports*).

The ImageNet-project [3] aims to populate the majority of the 80,000 synsets of WordNet [9] with an average of 500-1000 clean and full resolution images. By all standards, this is an impressive amount of concepts and an impressive amount of examples per concept. We turn our attention to the hierarchical composition of concepts, as for these or even larger numbers of concepts beyond these huge numbers, the hierarchical composition is important. We consider *groupings* or *categories* of objects. There are not only *cars*, *buses* and *bikes*, but also *vehicles*. And there are not only *tigers*, *dogs* and *hyena's*, but also *carnivores*, *vertebrates* and *animals*. An illustration of this hierarchy is shown in figure 1.

We follow the structure of WordNet which organizes English concepts in a hierarchy, using `is-a` relationships. In the WordNet hierarchy, the *basic* concepts are the leaf nodes of the WordNet tree. The non-leaf nodes are indicated as *container* concepts. Of course, different domains require different levels of distinction: when finding different types of fruit, it might suffice to distinguish *apples* from *oranges*, making *apple* the basic concept. However, in an automated grocery store checkout application, it is necessary to distinguish a *Jonagold* from a *Granny Smith*, and *apple* turns into

ICMR '12, June 5-8, Hong Kong, China

[1] See `http://www.oed.com/public/about`.



(a) vehicles

(b) carnivores

Figure 2: Sample of pictures from the ImageNet dataset annotated with container concepts. For both concepts, there is a strong preference for a specific descendant of the container concept.

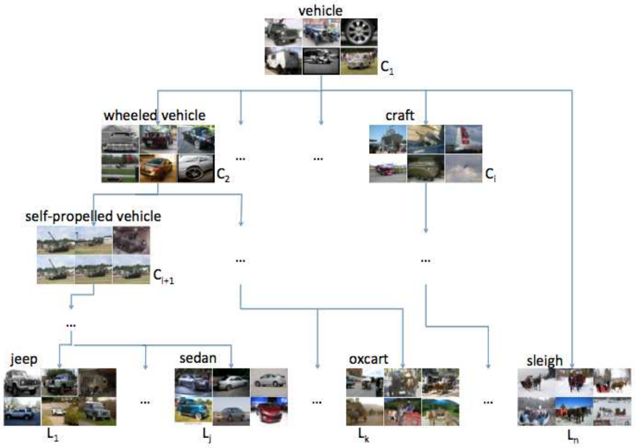


Figure 1: A selected view of the subset of the ImageNet hierarchy used for experiments in this paper.

a container concept.

Mathematically, a container concept is the union over all subcategories. However, the composition of ImageNet indicates that the contribution of descendants to the container is not uniformly weighed. In figure 2(a), a random subset of images annotated with *vehicle* is shown. It is clear that there is a preference in the representation. The vast majority of *vehicle*-images are *cars* and buses. The same holds for the set of images annotated with *carnivore* in figure 2(b): the majority of those images contain big cats. This may be a faithful representation of the a priori probability of encountering a picture of a big cat relative to other members of the carnivore class, but there are a few observations to be made here. One, the number of pictures of carnivores in the world is not a closed set, so a priori probability and fair representation may become meaningless quantities. Two, when learning to recognize concepts, an equal distribution capturing all variation is more important than a fair distribution. As even the variation in the world is endlessly complex, it is difficult for open classes to find a fixed recipe how to sample and how many to sample. And, finally, we argue that pictures ideally suited to cover the *vehicle* class are identical in their appearance to either the *car* or *bicycle*. When selecting samples, not every car is a vehicle, and surely in common language, a bicycle is less of a vehicle than a car is. In the paper we will examine the consequence of each of these observations.

As indicated in [3], the first step in populating a synset in ImageNet is querying several image search engines with the set of WordNet synonyms for that synset, and subsequently using Amazon Mechanical Turk to let humans judge the quality of the candidate images. As a result, the set of candidate images for a synset only includes images that are put online with terms from the synset. This procedure, maybe the only practical procedure one has, biases the set of images under an ImageNet synset a strong indicator for the visual association humans have with the words of the synset.

As can be seen from figure 2, when humans think of a vehicle, they picture a car, less a boat or a missile, and when they think of a carnivore, they picture a big cat, and not an otter or a mongoose. In this paper, we aim to investigate whether container concepts generally exhibit such patterns. If so, this would indicate the usefulness of annotating at every level of the hierarchy especially when selecting data for learning how to recognize container concepts.

In this paper, we aim to answer two main research questions:

Q 1. *Is a container concept classifier trained from examples at that level of abstraction visually different from a classifer trained on the examples of all leaf nodes under that container concept?*

Q 2. *Is a container concept detector when learned from lumping all examples at leaf nodes different from one learned from the a posteriori results of leaf node concept detectors?*

In this experimental study we describe the occurrence of preferred classes for container concepts in details, and we show the performance when training and testing on different datasets describing container concepts.

2. RELATED WORK

One of the greatest challenges in dealing with huge datasets is computational: when the amount of images runs in the millions, it becomes unfeasible to train concept detectors on general hardware. One of the angles to tackle this problem is to exploit the natural ordering present in concepts [3, 1].

In this paper, a subset of the ImageNet dataset is used, which was introduced in [3]. Here it was shown that exploiting the ImageNet hierarchy can provide substantial improvement for the image classification task. However, these results did not take into account any human preference for specific subclass images when labeling or searching for images annotated with container concepts: any image lower in the ImageNet hierarchy than a certain container concept was considered a positive example for that concept.

The strength of exploiting hierarchical knowledge was investigated further in Deng et. al. [1] for the problem of similar image retrieval. In this work, a similarity function was defined using the ImageNet hierarchy, and it was shown that adding hierarchical knowledge based on this function significantly increased retrieval performance. The similarity between two concepts was defined in terms of the lowest common ancestor of these concepts, however also without taking any preference effect for that ancestor concept towards any of its descendants into account.

Deng et. al. [2] demonstrated that a correlation exists between the structure of the semantic hierarchy and visual confusion between the categories, i.e. neighboring concepts in the ImageNet hierarchy are (broadly) more likely to be confused with each other than with concepts further away in the hierarchy. In the current work, the relation between concepts in a specific relation (namely ancestor-descendant) is investigated more closely. Thus, we do not explore the relationship between neighboring concepts (like e.g. *car* vs. *bus*), but instead focus on the relation between *car* and *vehicle*.

Rohrbach et. al. [8] also investigated the relation between container concepts and their constituting leaf nodes on the ImageNet Large Scale Visual Recogntion Challenge (ILSVRC). There, it was shown that using leaf nodes as positive examples for a container class gave better results when testing on the leaf nodes than using the images annotated for that container concept. In our work, we test not only on the leaf nodes, but also invesigate the classification performance on images annotated with the container concepts. We also provide qualitative analysis of the classifiers created from the different training sets.

3. EXPERIMENTAL SETUP

3.1 Dataset

The starting point for the experiments is the 2011. This is the largest image classification competition to date, using a dataset consisting of 1.23 million training images which are hand labeled for the presence of 1,000 object categories. These images are a subset of the ImageNet dataset. The 1,000 categories do not overlap.

Following the setup of the ILSVRC, the child categories of the 1,000 object categories were (if present) not considered. Three concepts positioned high in the ImageNet hierarchy were selected as subtree roots, namely *musical instrument*, *vehicle*, and *carnivore*. The images corresponding to these nodes, the 1,000 leaf nodes, and all concepts in between (131 in total) were downloaded from the ImageNet database. For each concept, at most 1,300 images were used as positive training examples (less than 1,300 only when not enough images were available). An illustration of a part of this hierarchy is shown in figure 1.

The validation set from the 2011 ILSVRC challenge was used as a test set. This test set consists of 50 images for each of the 1,000 concepts, leading to 50,000 images in total. This set will be indicated by T_L. As a second test set, from the downloaded container concepts images that contain more than 1,300 images, at most 50 images per concept are selected. This gives a total of 2,292 additional test images, which will be indicated by T_C.

3.2 Implementation Details

All classifiers were trained in a one-vs-all setting, using 1,300 positive examples, and taking examples from all other classes as negatives. This gives over 1.2 million negatives to choose from for each class. To reduce the computational load, we selected a fixed random subset of 10% (120,000 images) that were used as negatives for each experiment. When training a classifier for a concept, all images annotated with that concept or a descendant of it in the hierarchy were removed from the set of negative examples.

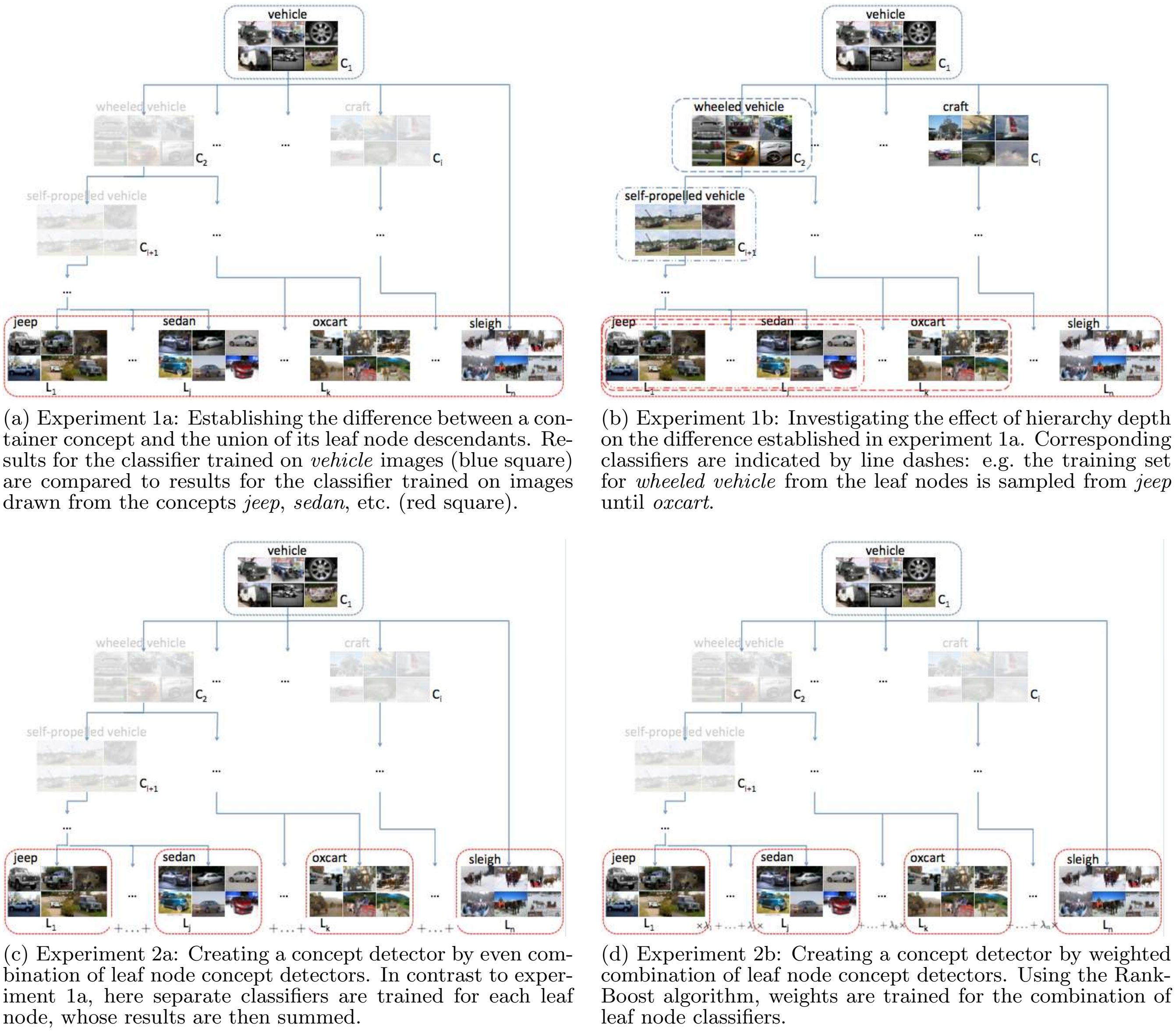
Bag of Words Images are represented using the standard bag-of-words model, with the SIFT [6], OpponentSIFT and RGB-SIFT descriptors extracted using the Color Descriptor software from [10]. The descriptors are extracted at Harris-Laplace keypoints and densely sampled every 6 pixels at two scales. The codebook size is 4096 and there are 1x3 spatial pyramid subdivisions [5]. As classifier we employ a Support Vector Machine with a histogram intersection kernel. We use the fast, approximate classification strategy of [7].

3.3 Experiments

The experimental setup is illustrated in figure 3.

Experiment 1a: Container vs. union of constituents To find the answers to research question 1, the first experiment establishes a difference between a classifier trained on the images labeled with a container concept, indicated by D_C, and one trained on images labeled with any of the leaf nodes, indicated by D_L. This experiment is performed on the four root nodes mentioned indicated in section 3.1, and is illustrated in figure 3(a). When training the classifier using the images labeled with the leaf nodes, a random selection of all available images is used, such that the amount of training examples for both classifiers is equal. Results are calculated on both test sets T_L and T_C.

Experiment 1b: Effects of hierarchy depth We expect that the differences between a container class and a union of its descendants, as established in experiment 1a, become larger with tree depth, i.e. when the distance in the tree between a container concept and its constituting leaf node becomes larger. For each container concept between the selected root nodes and the 1,000 leaf nodes, pairs of classifiers were trained in the same way as in experiment 1. This experiment is illustrated in figure 3(b). Results for both classifiers are obtained on test set T_L and the differ-



(a) Experiment 1a: Establishing the difference between a container concept and the union of its leaf node descendants. Results for the classifier trained on *vehicle* images (blue square) are compared to results for the classifier trained on images drawn from the concepts *jeep*, *sedan*, etc. (red square).

(b) Experiment 1b: Investigating the effect of hierarchy depth on the difference established in experiment 1a. Corresponding classifiers are indicated by line dashes: e.g. the training set for *wheeled vehicle* from the leaf nodes is sampled from *jeep* until *oxcart*.

(c) Experiment 2a: Creating a concept detector by even combination of leaf node concept detectors. In contrast to experiment 1a, here separate classifiers are trained for each leaf node, whose results are then summed.

(d) Experiment 2b: Creating a concept detector by weighted combination of leaf node concept detectors. Using the RankBoost algorithm, weights are trained for the combination of leaf node classifiers.

Figure 3: Overview of the experimental setup. Transparent parts in the subfigures are not used for that experiment. A square in a figure indicates that in the indicated experiment, a classifier is trained on images from the concept or concepts within the square. Blue squares indicate to classifiers trained on images labeled with container concepts (D_C), red squares indicate to classifiers trained on images sampled from leaf node images (D_L). Squares with the same dash type in one subfigure indicate that corresponding classifiers are trained from those sets.

ence in AP between the classifier as a function of the average distance between the container concept and its leaf node descendants is used for evaluation.

Experiment 2a: Combining constituent classifiers Another way of creating a high-level concept classifier is to combine the outputs of classifiers for its constituents. For this, the outputs of the leaf node classifiers $D_{L_1} \dots D_{L_n}$ are simply summed to create a classifier for the container concept C of which $L_1 \dots L_n$ are the leaf node descendants. The detector for container concept C which is thus obtained is indicated by $D_{L_1 \dots L_n}$. This experiment in illustrated in figure 3(c). Results for this experiment are again evaluated on both test sets T_L and T_C.

Experiment 2b: Weighted subclass combination The occurrence of preferred subclasses suggests that better results for the combination of subclass detectors can be obtained when attaching weights to each concept detector output. This experiment in illustrated in figure 3(d).

For this experiment, each leaf node concept detector below a container is interpreted as a weak classifier for that container concept. The RankBoost algorithm as implemented

concept	classifier	T_C	T_L
musical instrument	D_C	0.10	0.18
	D_L	0.09	0.20
vehicle	D_C	0.39	0.49
	D_L	0.11	0.59
carnivore	D_C	0.35	0.39
	D_L	0.05	0.51
MAP	D_C	0.40	0.20
	D_L	0.28	0.39

Table 1: Classification performance in average precision of experiment 1a. T_L and T_C indicate the test sets of leaf node images and container images respectively. D_C and D_L indicate classifiers trained on images labeled with the container concept and with a sample from the leaf nodes respectively. Note that Mean Average Precision (MAP) is calculated over *all* container nodes, not just the three root nodes shown here.

by Van Dang[2] was used to learn how best to combine the leaf node classifiers. From the training images for each of the container concepts, 50 images were selected on which the boosting algorithm was trained. The resulting classifier is indicated by $\hat{D}_{L_1 \dots L_n}$ and its performance was compared with the classifiers trained directly on the container images and with the unweighted classifier combination from experiment 2a on test set T_C.

4. RESULTS

Experiment 1a: Container vs. union of constituents. In table 1, the results on the two test sets are presented. It can be observed that a detector trained on images labeled at the container level performs better on test images also labeled at the container level, while a detector trained on images labeled at the leaf node level performs better on test images also labeled at the leaf node level. Furthermore, the differences in classification performance for the detectors trained on the two different image sets are quite large for the *vehicle* and the *carnivore* concepts (with differences of 20% and 31% relative on test set T_C respectively, and differences of 72% and 86% relative on test set T_L respectively).

In figure 4, the 35 highest ranking output images for the two different training sets for *vehicle* are shown. It can be seen that for the classifier trained on images labeled as *vehicle* (figure 4(a)), all of the top-35 images depict cars, while for the classifier trained on images labeled as any of vehicles' leaf node descendants (figure 4(b)), the top-35 images depict several different vehicle-types (*boats*, *bikes*, and *cars*). Thus, the first classifier gives an overview of the images that humans generally label as *vehicle*, while the second gives an overview of the images that *could* be labeled as *vehicle*.

Experiment 1b: Effects of hierarchy depth. The results for experiment 1b are shown in figure 6. Recall that we hypothesized that the performance difference between a classifier trained on images labeled with the container concept and one trained on images sampled from its constituting leaf nodes would grow smaller with the average distance between the container and the constituting leaf nodes. As can be seen from the figure, this is not the case. Note that many of the container concepts have an average distance to their leaf node descendants of 1 (meaning they are the direct parents of their constituting leaf nodes), but the differences in AP between the two classifiers for those concepts range from −0.09 (the negative indicating that the classifier D_C outperforms D_L on test set T_L) to 0.75. Finally, note that almost all of the data points are above 0, indicating that on test set T_L (i.e. the test set sampled from images annotated with leaf node labels), the classifier trained on the images labelled with leaf node labels almost always outperforms the classifier trained on the images labelled with container node labels. This confirms the impression from table 1. Additional experiments have confirmed that on test set T_C, classifiers D_C outperform D_L (data not shown).

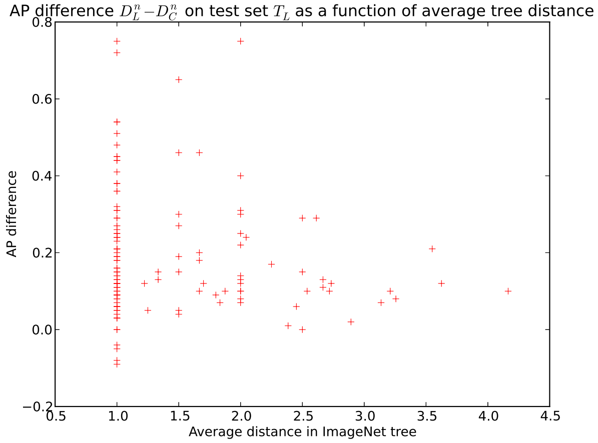
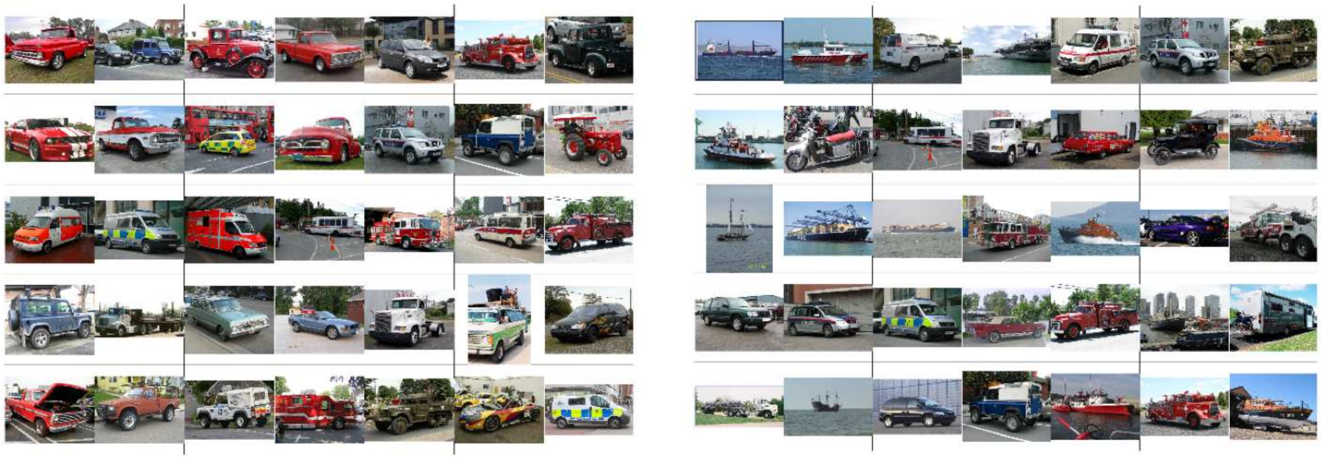


Figure 6: Results for experiment 1b. On the x-axis, the average distance between a container concept and its leaf node descendants is plotted, and on the y-axis, the difference (in AP) between the two concept detectors trained on images labeled with the container concept and with its leaf node descendants, when tested on leaf-node images. Each data point represents a container concept. The plot shows that the performance difference between the two classifiers is independent of the height of the container concept.

An illustration of the fact that a classifier trained on a container element closer to its constituting leaf nodes can differ more is given in figure 5, which shows the top 35 results for the concept *self-propelled vehicle* for the two different classifiers. *Self-propelled vehicle* has an average distance to its leaf node descendants of 2.6, while its ancestor *vehicle* has an average distance of 4.2. In contrast, the difference in AP between the classifiers trained on the container images and on the constituting leaf node images is 0.29 and 0.1 respectively. The reason for this large difference in performance can be seen in figure 5(a): almost all images in *self-propelled vehicle* are of military vehicles, while there are no images of military vehicles in the 1,000 concepts of the ILSVRC dataset. Note that in the ImageNet hierarchy, there are military vehicles as descendants of the self-propelled vehicle class. These are, however, a small minority of all its descendants. When repeating the experiment using all constituting

[2] http://www.cs.umass.edu/∼vdang/ranklib.html



(a) Results for classifier trained on container-labeled images.

(b) Results for classifier trained on leaf node-labeled images.

Figure 4: Top 35 results for two different *vehicle* classifiers from experiment 1a. Note that the classifier trained on images labeled as *vehicle* has only cars as its highest-ranking output images, but the one trained on the leaf node images has much more diverse high-ranking images.

concept	classifier	T_C	T_L
musical instrument	D_L	0.09	0.20
	$D_{L_1 \ldots L_n}$	0.09	0.44
vehicle	D_L	0.11	0.59
	$D_{L_1 \ldots L_n}$	0.13	0.75
carnivore	D_L	0.05	0.51
	$D_{L_1 \ldots L_n}$	0.06	0.69
MAP	D_C	0.40	0.39
	D_L	0.29	0.44

Table 2: Classification performance in average precision of experiment 2a. T_L and T_C indicate the test sets of leaf node images and container images respectively. D_C and $D_{L_1 \ldots L_n}$ indicate classifiers trained on images labeled with the container concept and by combining classifiers from the constituting leaf nodes respectively. Note that Mean Average Precision is calculated over *all* container nodes, not just the three root nodes shown here.

concept	classifier	T_C
musical instrument	D_C	0.10
	$D_{L_1 \ldots L_n}$	0.09
	$\hat{D}_{L_1 \ldots L_n}$	0.08
vehicle	D_C	0.39
	$D_{L_1 \ldots L_n}$	0.13
	$\hat{D}_{L_1 \ldots L_n}$	0.26
carnivore	D_C	0.35
	$D_{L_1 \ldots L_n}$	0.06
	$\hat{D}_{L_1 \ldots L_n}$	0.18

Table 3: Classification performance in average precision of experiment 2b. Results are calculated on test set T_C, consisting of images annotated with the container node labels. images respectively. D_C, $D_{L_1 \ldots L_n}$, $D_{L_1 \ldots L_n}$ indicate classifiers trained on images labeled with the container concept, as combination of constituting leaf node classifiers, and as weighted combination of constituting leaf node classifiers respectively.

leaf nodes of *self-propelled vehicle*, the difference is expected to decrease slightly. However, the imbalance between the amount of military vehicles in *self-propelled vehicle* and in its constituting leaf nodes will remain significant.

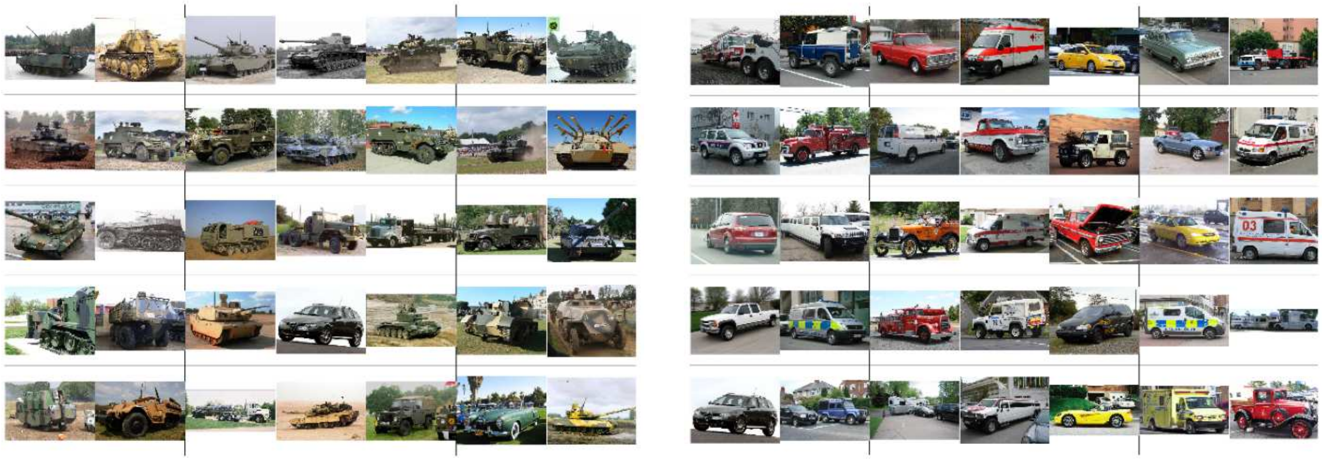
Experiment 2a: Combining constituent classifiers. Table 2 presents the results for the concept detectors trained by combining leaf node detectors on test sets T_L and T_C. For comparison, the results for the detectors that were sampled from all leaf nodes (D_L) are also given.

As can be seen from the table, the detector created by the summation of the leaf node detectors ($D_{L_1 \ldots L_n}$) outperforms the basic container level detector when testing on the images labeled with leaf nodes (T_L). The differences in performance are significantly larger than in experiment 1a. This can be explained by the fact that each leaf node classifier is trained specifically for the images on the same level as the test set. Furthermore, since each classifier is trained with the same amount of training images, the combined classier created here benefits from a much larger total set of training examples (namely $n \times 1,300$), which always increases classifier performance.

When testing on images labeled at the container level, there is only a slight difference in performance between combining leaf node detectors, and training on a sample from all leaf nodes. This shows that both methods are not equipped to capture the preference phenomenon that we have seen so far. These results also extend to the lower level container nodes used in experiment 1a (data not shown).

Experiment 2b: Weighted subclass combination. Using the RankBoost algorithm, we trained a boosting classifier with a selection of the container-labeled training images as ground truth for the boosting algorithm. Results are shown in table 3.

The table shows that for carnivore and vehicle, a large performance gain can be achieved over the unweighted com-



(a) Results for classifier trained on container-labeled images.

(b) Results for classifier trained on leaf node-labeled images.

Figure 5: Top 35 results for two different *self-propelled vehicle* classifiers from experiment 1b. Note that the images in (a) differ even more from those in (b) than the images in figure 4(a) differ from those in 4(b), even though in the ImageNet hierarchy, *self-propelled vehicle* is on average closer to its constituting leaf nodes than *vehicle*.

bination of leaf node classifiers. However, on the full set of container concepts, the weighted subset combination in fact performs worse than the unweighted version, with large differences between cocnepts (data not shown). This indicates that care needs to be taken when using this type of two-step training: with too little or poorly chosen training data, results can be in fact harmful to performance.

5. CONCLUSIONS AND DISCUSSION

In this paper, we have shown that the set of images annotated with a container concept is substantially different from the union of the sets of images annotated with its leaf nodes. By training classifiers on images annotated with a container concept and comparing them to classifiers trained on their constituting leaf node concepts, we have shown that the answer to our first research question is affirmative: there exists a strong visual difference between these two classifiers. Furthermore, we have shown that this difference does not necessarily decreases when the container concept covers less diverse leaf node concepts.

Secondly, we have shown that while training separate classifiers for the leaf node concepts improves performance when testing on leaf node-labeled images, performance on container-node images stays generally similar when testing on images labeled with the container concept. When training a boosting algorithm on the container-labeled training images, performance does improve. This indicates that the preference problem cannot be solved by simply using more training data, the actual class preferences need to be considered.

This study confirms the bias effect that exists in images on the web: images annotated with a container concept are not evenly distributed amongst the basic concepts that make up the container. This indicates that, when creating an image classification system for such container concept, careful consideration about the desired behavior is necessary: When the aim is to simply rank images based on whether they contain e.g. *any* type of vehicle, using images from all concepts below *vehicle* as training examples are useful. However, if the aim is to adequately reflect the bias properties of the container concept, only images explicitly labeled as *vehicle* should be used as training examples.

6. REFERENCES

[1] J. Deng, A. Berg, and L. Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *CVPR 2011*.

[2] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10, 000 image categories tell us? In *ECCV 2010*.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*.

[4] R. Goulden, P. Nation, and J. Read. How large can a receptive vocabulary be? *Applied Linguistics*, 11(4), 1990.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006*.

[6] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV 2004*.

[7] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR 2008*.

[8] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR 2011*.

[9] M. Stark and R. Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*, 1998.

[10] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI 2010*.