How to Transfer? Zero-Shot Object Recognition via Hierarchical Transfer of Semantic Attributes

Ziad Al-Halah   Rainer Stiefelhagen

Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology

{ziad.al-halah, rainer.stiefelhagen}@kit.edu

Abstract

Attribute based knowledge transfer has proven very successful in visual object analysis and learning previously unseen classes. However, the common approach learns and transfers attributes without taking into consideration the embedded structure between the categories in the source set. Such information provides important cues on the intra-attribute variations. We propose to capture these variations in a hierarchical model that expands the knowledge source with additional abstraction levels of attributes. We also provide a novel transfer approach that can choose the appropriate attributes to be shared with an unseen class. We evaluate our approach on three public datasets: aPascal, Animals with Attributes and CUB-200-2011 Birds. The experiments demonstrate the effectiveness of our model with significant improvement over state-of-the-art.

1. Introduction

Semantic attributes describe the object's shape, texture and parts. They have the unique property of being both machine detectable and human understandable. By changing the recognition task from labeling to describing, attributes represent an adequate knowledge that can be easily transferred and shared with new visual concepts. Thus, they can be used to recognize unseen classes with no training samples (*i.e.* zero-shot classification).

In the prevailing approach, attributes are learned from all seen classes and then reused to describe or classify an unseen one. However, this doesn't account for the high intra-attribute variance. Using all the seen classes helps in learning visual semantics in a very abstract manner. Hence, subsets of classes that share similar attributes cannot be distinguished easily. Eventually, the fine properties of the attribute that help in discriminating a group from another are lost when it is learned from all the classes. Consider for example the attribute *beak* in Figure 1. The global attribute model would learn that a beak is an elongated extension at a certain position relative to the head; *i.e.* ignoring the dis-
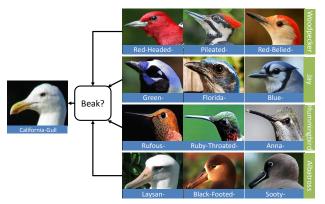

Figure 1: The high intra-attribute variance is better represented at different semantic levels of abstraction. This helps in directing the transfer process to identify the most suitable source of knowledge to share with a novel class.

tinctive long thin beak shape of the hummingbird species or the wide curved-end of the albatross species. In other words, the global model does not take advantage of the rich information already available in the source dataset. This results in transferring less discriminative attributes to the novel class. On the other hand, capturing these specific properties of *beak* relative to each subgroup of birds is beneficial. It gives us the option to select the most proper type of *beak* to share with the unseen class. Accordingly, knowing that both *Gull* and *Albatross* are *Seabirds*, it is intuitive and probably more discriminative to describe the beak of the *California-Gull* as an *albatross-like-beak*.

In this work we study the benefit of learning attributes at different levels of abstraction, from the most specific that distinguish one class from another to the most general that are learned over all categories. We propose a novel hierarchical transfer model that can find the best type of attributes to be shared with an unseen class. We evaluate the proposed model on three challenging datasets each with a different granularity of object categories. The evaluation shows that significant gain can be achieved with our guided transfer model with improvements from 22% and up to 32% over state-of-the-art in zero-shot classification.

2. Related work

Our work relates to two fields in computer vision literature; the attribute-based recognition and hierarchical transfer learning.

Since the introduction of semantic attributes [10, 9, 14], they have gained increasing attention from the computer vision community. They represent an intermediate layer of semantics and facilitate various tasks in visual recognition like zero-shot learning [9, 8, 14], aiding object classification and localization [24], relative attribute comparison [18] and detection of unfamiliar classes [22]. However, in the prevailing direction the attributes are learned in a global manner from all classes available in the source [14, 18, 9, 16]. Such methods cannot cope with the high variations in each of the attributes. Some approaches jointly model objects, attributes and their correlations [24, 16] to handle such variations. Nonetheless, these correlations cannot be learned for unseen classes since there is no training data.

Our approach is related to the work on learning class-specific attributes. In [9] a set of attributes are learned per class as an intermediate step for feature selection in order to reduce attribute correlations. Yu *et al.* [25] propose to learn data-driven attributes at the category-level to better discriminate the classes. However, data-driven attributes usually carry no semantic meaning; thus, their approach requires user interaction when performing zero-shot learning. In [27] the concepts in ImageNet are augmented with a set of semantic and data-driven attributes. These are used along with the hierarchy to learn a better similarity metric for content-based image retrieval. Correspondingly, we propose to explicitly model the intra-attribute variations at different abstraction levels. That is, rather than just using class-specific attributes, we expand the notion of the attribute from the most abstract to the most specific driven by the embedded relations between the categories.

Additionally, hierarchies represent an attractive structure for knowledge transfer and they have been exploited in various ways: parameter transfer [21, 20], representation transfer [2] and bounding box annotations propagation [12]. Of particular relevance to our work is the joint modeling of hierarchy and attributes. In [1], a ranking classifier is trained using attributes for label embedding; showing that using hierarchical labels along global attributes as side information improved the zero-shot performance. In the recent work of [4], a hierarchy and exclusion graph is learned over the various object categories. The graph models binary relations among the classes like mutual exclusion and overlap. They also model similar relations between objects and global attributes and use it for zero-shot classification.

We exploit the hierarchical structure of the categories in two aspects. We leverage the hierarchy to automatically propagate annotations and learn attributes at different levels of abstraction. Then we use it in guiding the transfer process to select the most promising knowledge source of attributes to share with novel classes. To the best of our knowledge, this has not been done before.

3. Approach

Hierarchical representation of concepts and objects is part of the human understanding of the surrounding world. We usually try to combine objects into certain groups based on a common criteria like functionality or visual similarity. This helps us to better learn the commonality as well as the differences in and across groups. The key idea of our approach is to take advantage of the embedded structure in the object category space and extend the notion of global attributes to include different levels of abstraction. The object hierarchy groups the classes based on their overall visual similarity; thus provides a natural way to guide the transfer process to share information from the knowledge sources that will most likely contain relative information.

In the following, we describe the three main steps of our method. Starting with a set of classes, global attributes and a hierarchy in the category space: (1) we automatically populate the hierarchy with additional attributes; (2) learn these attributes to capture subtle differences between similar categories; and finally, (3) use a hierarchy-guided transfer to select the proper attributes to share with a novel class. We start by defining the notation used throughout the paper.

Notation. Let $\mathcal{C} = \{c_k\}_{k=1}^K$ be a set of categories, where a subset of these categories $\mathcal{Q} = \{q_t\}_{t=1}^T$ have training samples while the rest $\mathcal{Z} = \{z_l\}_{l=1}^L$ have none, and $\mathcal{C} = \mathcal{Q} \cup \mathcal{Z}$. A set of semantic attributes $\mathcal{A} = \{a_m\}_{m=1}^M$ describe all classes in $\mathcal{C}$. A directed acyclic graph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ defines a hierarchy over the classes, with nodes $\mathcal{N} = \{n_i\}_{i=1}^I$ and edges $\mathcal{E} = \{e_{ij} : n_i, n_j \in \mathcal{N}\}$. An attribute a_m at node n_i is referred to as $a_m^{n_i}$.

3.1. Populating the hierarchy with attributes

To get the attribute description of the inner nodes we exploit the hierarchy $\mathcal{H}$ by transferring the attributes annotation in a bottom-up approach from the seen classes $\mathcal{Q}$ (leaf nodes) to the root. For example, the inner node *Dog* (Figure 2a) has the attributes *patches* leveraged from the child nodes *Collie* and *Dalmatian*, which in turn propagates *patches* along other attributes up to node *Carnivore*. Then, the active attributes of node n_j are defined as

$$a_m^{n_j} = 1 \text{ if } \exists a_m^{n_i} = 1 \text{ and } n_i \in \text{child}(n_j), \qquad (1)$$

where $\text{child}(n)$ is the set of nodes of the subtree rooted with n. Consequently, the root node of $\mathcal{H}$ will be described with all attributes of $\mathcal{Q}$. It's important to note that the attribute label propagation is used to find the active attributes at a certain node and to guide the transfer process afterwards. This does not change the underlying ground truth attribute
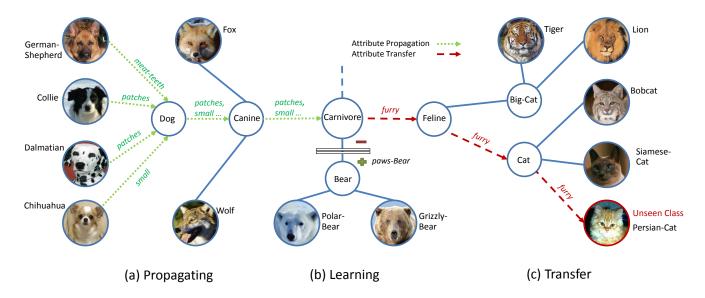
Figure 2: Illustrative figure of our hierarchical transfer model. We leverage the hierarchy in many aspects: (a) attribute label propagation, (b) modeling the intra-attribute variations and (c) guiding the transfer process to select and share the most relevant knowledge source for a novel class. (Best viewed in color)

labels of samples whether it is class-based or image-based annotation.

3.2. Learning at different levels of abstraction

To learn the various attributes classifiers, we first define the support set of an attribute a_m, *i.e.* the set of samples that provide evidence of a_m. An attribute $a_m^{n_j}$ in the hierarchy has the support set $\text{supp}(a_m^{n_j})$. The set contains samples labeled with the attribute of that class ($\text{lbl}(a_m^{n_j}) : n_j \in \mathcal{Q}$), and additionally the samples of its children which share the same attribute with n_j, *i.e.*

$$\text{supp}(a_m^{n_j}) = \bigcup_{n_i \in \text{child}(n_j)} \text{supp}(a_m^{n_i}) \cup \text{lbl}(a_m^{n_j}). \quad (2)$$

To capture the fine differences that characterize an attribute at node n, we use a child-vs-parent learning scheme [17]. The attribute $a_m^{n_c}$ is learned with the following positive (T_P) and negative (T_N) sets

$$T_P = \text{supp}(a_m^{n_c}) \quad T_N = \text{supp}(a_m^{n_p}) - \text{supp}(a_m^{n_c}), \quad (3)$$

where n_p is the parent node of n_c. For example, the attribute *paws* of node *Bear* (Figure 2b) is supported by *paws* samples from the classes *Polar-Bear* and *Grizzly-Bear*. Then, a_{paws}^{Bear} is learned to capture the differences against the other *paws* samples from parent *Carnivore*. The root n_r of $\mathcal{H}$ has no parent; hence $\{a_m^{n_r}\}$ are learned in the standard 1-vs-all scheme. In other words, the root attributes naturally map to the commonly used global attributes in the literature.

3.3. Hierarchical transfer

Having the attributes learned at different levels of abstraction, we then leverage the hierarchy to guide the knowledge transfer process and find the proper attributes to transfer to novel classes.

Analyzing the recall and precision properties of the attribute classifiers in $\mathcal{H}$, we generally notice that the attribute predictions towards the leaf level of $\mathcal{H}$ have higher precision and lower recall than the ones towards the root of the hierarchy, which are characterized with low precision and high recall. This is expected from a learning scheme as the one we use. While the lower levels capture the discriminative small differences that distinguish a small group of classes against another regarding an attribute (high precision), in the higher levels of $\mathcal{H}$, the common visual properties of the attribute across a larger set of categories is learned (high recall). Similar to [28], we find that combining classifiers with such an opposite recall and precision primacy results in an improved performance when compared to the constituent classifiers. Furthermore, the combination of classifiers from different levels in the hierarchy increases the robustness of the final classifier against noise that might be produced by the constituent classifiers.

Accordingly, for a novel class z_l in $\mathcal{H}$, we transfer the attributes of its ancestors across the different levels of abstraction (*e.g.* $a_{furry}^{PersianCat}$ in Figure 2c), such that:

$$s_{z_l}(a_m^{z_l}|\mathbf{x}) = \frac{\sum_{n_i \in \text{anc}(z_l)} [[a_m^{z_l} = a_m^{n_i}]] \, s_{n_i}(a_m^{n_i}|\mathbf{x})}{\sum_{n_i \in \text{anc}(z_l)} [[a_m^{z_l} = a_m^{n_i}]]}, \quad (4)$$

where $[[\cdot]]$ is the Iverson bracket (*i.e.* $[[P]] = 1$ if condition P is true and 0 otherwise), $s_n(a_m^n|\mathbf{x})$ is the score of the attribute a_m for node n given sample $\mathbf{x}$, and $\text{anc}(n)$ is the set of ancestor nodes of n. Once the attributes are transferred to z_l, the final prediction score $s(z_l|\mathbf{x})$ of the z_l category can be defined by averaging over the attributes of that class as:

$$s(z_l|\mathbf{x}) = \frac{\sum\limits_{m=1}^{M} [[a_m^{z_l} = 1]]\, s(a_m^{z_l}|\mathbf{x})}{\sum\limits_{m=1}^{M} [[a_m^{z_l} = 1]]}. \tag{5}$$

In the following, we refer to our Hierarchical Attribute Transfer model as HAT.

4. Evaluation

We evaluate our model using three datasets. Each provides different characteristics regarding the granularity of classes. This give us the chance to see how the performance of the proposed HAT model varies with regards to the complexity of the embedded knowledge in the source set. Next, we present the three datasets, the hierarchy and the features we extract from images to train the different attribute models.

Datasets. **(1)** The aPascal/aYahoo (aPaY) [9] contains two subsets. The first (aP) uses 12,695 images and 20 categories from PASCAL VOC 2008 [6]. The second (aY) has 12 disjoint classes and 2,644 images collected from the Yahoo image search engine. Per-image labels of 64 binary attributes are provided for both subsets. In the predefined zero-shot split, aP is used for training and aY for testing.

(2) The Animals with Attributes (AwA) [14] consists of 30,475 images of 50 classes of animals. They are described with 85 semantic attributes on the class level. The authors provide a fixed split for zero-shot experiments. They select 40 classes for training and 10 for testing.

(3) The CUB-200-2011 Birds (CUB) [23] has 200 bird classes and 11,788 images. Each image is labeled with 312 attributes. Unlike the previous two, there is no predefined split for this dataset. For our experiments, we randomly select 150 classes for training and 50 for testing.

Hierarchy. We learn the object hierarchies using the WordNet ontology. That is, by querying the ontology with the category labels we extract a tree that brings the classes into a hierarchical ordering. Subsequently, we prune the hierarchy to remove intermediate nodes that have a single child.

Deep Features. Motivated by the impressive success of deep Convolutional Neural Networks (CNN) [13], we encode the images using a CNN-based deep representation to train the attribute classifiers. We use the CNN model CNN-M2K provided by [3]. The CNN has 5 convolutional and 3 fully connected layers. The network is trained on ILSVRC-2012 from [5] for object classification. We follow the best

Features	aPaP	aPaY	AwA	CUB
Shallow	84.12	70.91	71.16	60.78
Deep	**92.82**	**80.73**	**78.64**	**76.03**

Table 1: Attribute prediction performance (mean AUC) using deep and shallow representations.

practice found in [3] to extract the deep representation. The image is resized to 256 x 256. Then 5 crops of the image at center and corners with their flipped versions are fed to the CNN. The output of the seventh layer is extracted, sum-pooled and L2-normalized to be used as our deep-features to train the different models[1]. For all our attribute classifiers, we use linear SVMs [7] with regularized logistic regression. The cost parameter C is estimated using 5-folds cross validation.

4.1. Attribute Prediction with Deep Features

We first evaluate the performance of the deep features that we use in learning attribute classifiers. We compare their performance with attributes learned using "shallow" features. We use the precomputed features provided by [9] and [14] as the "shallow" representation for aPaY and AwA respectively. For CUB, we encode the images with Fisher vectors based on SIFT and Color descriptors and use that as the shallow representation. In aPaY, we consider two setups: (i) within-category attribute prediction (aPaP), *i.e.* the evaluation is done on the aPascal testing set; (ii) across-category prediction (aPaY), *i.e.* we evaluate on the disjoint aYahoo set. In both cases, the attributes are learned using the aPascal training set. For AwA and CUB, we use the zero-shot testing setups defined in the previous section.

In Table 1, the performance of the two representations in terms of mean AUC under ROC of the attribute predictions is shown. The deep-feature model constantly outperforms its counterpart across all datasets with an increase between 7% to 15%. This high performance of the deep features in attribute prediction is expected. CNNs automatically learn to capture features with varying complexity at each layer. While the lower layers learn features like edges and color patches, the higher layers learn much complex structures of the object like parts [26]. Many of these correspond directly to semantic attributes which explains the impressive performance of the deep representation using our simple linear classifiers.

4.2. Zero-Shot Classification

To populate the hierarchy with attributes (Section 3.1), our model requires class-based attribute descriptions. Hence, for aPaY and CUB we average all attribute vectors

---

[1]The deep features used in this work will be available on our website: http://cvhci.anthropomatik.kit.edu

Model	Features	aPaY	AwA	CUB
DAP [15]	shallow	19.1	41.4	-
IAP [15]	shallow	16.9	42.2	-
AHLE [1]	shallow	-	43.5	[17.0] [2]
HEX [4]	deep	-	44.2	-
TMV-BLP [11]	shallow	-	47.1	-
DAP	deep	31.9	54.0	33.7
ENS	deep	31.7	57.4	29.0
HAT (ours)	deep	**38.3**	**63.1**	**44.4**
ENS-n	deep	43.1	57.7	37.3
HAT-n (ours)	deep	**46.3**	**68.8**	**48.6**

Table 2: Zero-shot multi-class accuracy on the three datasets.

Model	Features	aPaY	AwA	CUB
DAP	deep	**87.3**	88.5	82.2
ENS	deep	85.6	88.5	89.5
HAT (ours)	deep	87.1	**92.0**	**94.9**

Table 3: Zero-shot mean AUC under ROC curve for the test classes.

of each class to calculate the class-attribute occurrence matrices. Then, the binary class-attribute notations are created by thresholding the resulting occurrence matrix at its overall mean value. Along with our Hierarchical Attribute Transfer model (HAT), we also train two common baselines for global attributes using our deep features: (i) The Direct Attribute Prediction model (DAP), where the class prediction is formulated as a MAP estimation [14]; (ii) The Ensemble model (ENS), that combines the predictions of the attributes using a sum formulation similar to the one we use in Eq. 5 but based on global attributes [19].

HAT vs. state-of-the-art. In Table 2, we report the normalized multi-class accuracy on the three test sets. Our model outperforms the state-of-the-art on the three datasets with a wide margin. In [1], a model that uses object hierarchy as side information is used to learn attributes based on Fisher vectors. In the recent work of [4], a hierarchical model of objects is learned using deep features similar to the one we use. While in [11] a transductive multi-view embedding is learned using global attributes, word space and low-level features. Nonetheless, our model improves over the best state-of-the-art results by 19% (aPaY), 16% (AwA) and 27% (CUB).

HAT vs. deep-feature baseline. Compared to our strong baseline (DAP- and ENS-deep), HAT still performs the best in terms of both multi-class accuracy and mean AUC (Table 3). Figure 4 shows the highest ranking re-

---

[2]Although we followed a similar split on CUB as in [1], the actual classes may differ and the results may not be directly comparable.
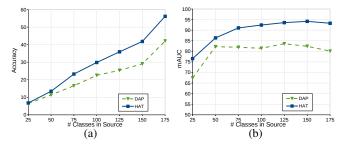

Figure 3: The performance of DAP and HAT in CUB with varying number of classes in the source as demonstrated with (a) multi-class accuracy and (b) mean AUC.

sults obtained by the three models for each test class in the AwA dataset. While distinctive classes like *Chimpanzee* and *Humpback-Whale* are correctly classified by all models, both DAP and ENS confuse visually similar classes that share many global attributes like *Leopard* & *Persian-Cat* and *Rat* & *Raccoon*. Consequently, more wrong samples are ranked high by these models. To the contrary, HAT learns the fine differences among the shared attributes of theses classes which helps in discriminating them efficiently. For example, HAT learns the differences between the attributes of *Big-Cat* and *Cat* (Figure 2c) which facilitates the separation among the novel classes *Leopard* and *Persian-Cat* (Figure 4).

Furthermore, we find that normalizing the prediction scores of the novel classes (Eq. 5) to have a zero mean and unit standard deviation makes the scores more comparable. This improves the accuracy of both the baseline (ENS-n) and our model (HAT-n) with the latter surpassing the former. However, this requires that the test data is available as a batch at the test time. The improvement in accuracy of HAT relative to the baseline is higher on AwA and CUB (19% and 30%) compared to aPaY (7%). This is expected since the classes in aPaY are visually farther apart from each other compared to the classes in AwA and CUB. Thus, it is harder for the baseline models (DAP & ENS) to distinguish such fine grained objects using the abstract global attributes.

Per-image vs. per-class attributes. aPaY and CUB provide attribute annotation at the image level which we used to train the models in the previous experiments. We evaluate on these two datasets using class-based attributes similar to those in AwA. We notice that the accuracy of both the baseline and HAT decreases in these settings. Where ENS has 22% and 29%, the HAT model achieves 33% and 40% on aPaY and CUB. This seems to differ from the findings in [15]. The reason could be related to the type of features used. In [15] a set of shallow features are used which require a relatively larger number of samples to train good attribute classifiers. This in turn results in noisy attribute predictions if there are few image-based annotations of the attribute. In comparison, using the deep features we can learn better attribute classifiers even if the training data is

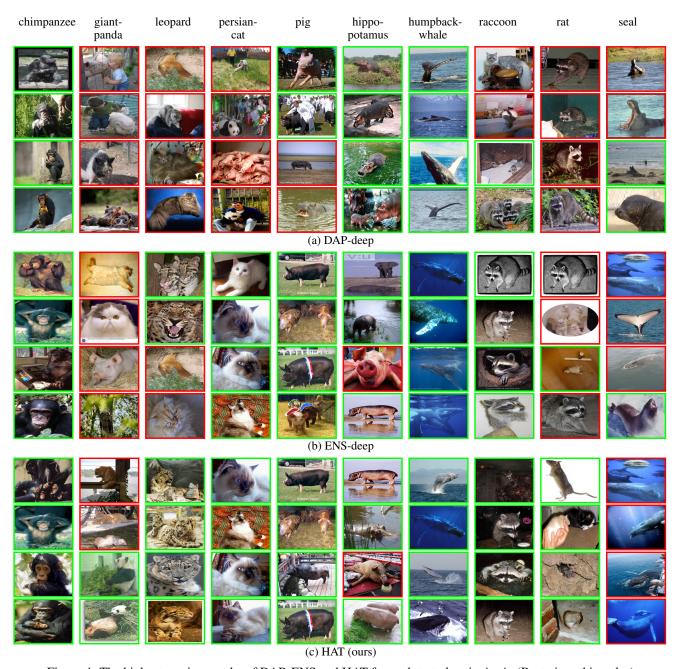| chimpanzee | giant-panda | leopard | persian-cat | pig | hippo-potamus | humpback-whale | raccoon | rat | seal |

(a) DAP-deep

(b) ENS-deep

(c) HAT (ours)

Figure 4: The highest scoring results of DAP, ENS and HAT for each test class in AwA. (Best viewed in color)

relatively sparse.

Unknown attributes of the novel class. Although this evaluation setup is not possible with the global attribute model, HAT enables us to carry out zero-shot recognition even if the attribute description of the novel class is unknown. To do that, we again leverage the hierarchy and transfer the attribute description of the parent node to the novel class. Using this setup, HAT achieves an accuracy of 21.1% (aPaY), 52.6% (AwA) and 25.9% (CUB). This drop in performance is reasonable since we are transferring the more generic attributes of the parent. Hence, confusion can

arise when multiple test classes share the same parent in the hierarchy. Nonetheless, HAT makes it possible to perform attribute-based zero-shot classification when only the novel class label is available.

Source set complexity. In the following experiment, we vary the complexity of the knowledge contained in the source (the number of seen classes) compared to the target (the unseen classes). This helps to have a better understanding of the characteristics of the different models as the richness of the embedded information in the source changes. We use the CUB dataset and start with a random set of 25

classes to be in the source. We gradually increase the source set with additional 25 random classes. At each step, the rest of the 200 classes is used as the target set to conduct zero-shot classification.

In Figure 3 we see that when the source is relatively poor and contains less structured knowledge, both DAP and HAT performs at the same level. However, as the source get bigger and more complex HAT consistently outperforms DAP with an increasingly wider margin. Unlike DAP that uses a single layer of global attributes, HAT is able to take advantage of the complexity of information available in the source. HAT captures the commonality among the categories and exploits it to learn and transfer more discriminative attributes to distinguish the unseen categories.

5. Conclusion

In this paper, we present a simple yet very effective model for zero-shot object recognition. Our model takes advantage of the embedded structure in the category space to learn attributes at different levels of abstraction. Furthermore, it exploits inter-class relations to provide a guided knowledge transfer approach that can select and transfer the expected relevant attributes to a novel class. The evaluation on three challenging datasets shows the superior performance of the proposed model over the state-of-the-art.

We are considering to extend our approach in different directions. In the current model, we assume similar importance for the transferred attributes from the different layers in the hierarchy. However, the confidence in the attributes predictions and their relative relatedness to the novel class differ through the different abstraction levels. Using adaptive weights when transferring attributes could improve the model performance. Moreover, we considered only the ancestor nodes as a knowledge source. The siblings of a novel class represent another promising option. Including them in the transfer process may help in sharing more discriminative attributes.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Attribute-Based Classification. In *CVPR*, 2013.

[2] Z. Al-Halah, L. Rybok, and R. Stiefelhagen. What to Transfer? High-Level Semantics in Transfer Metric Learning for Action Similarity. In *ICPR*, 2014.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *BMVC*, 2014.

[4] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-Scale Object Classification using Label Relation Graphs. In *ECCV*, 2014.

[5] 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.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, 2008.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[8] A. Farhadi, I. Endres, and D. Hoiem. Attribute-Centric Recognition for Cross-category Generalization. In *CVPR*, 2010.

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009.

[10] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *NIPS*, 2008.

[11] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive Multi-view Embedding for Zero-Shot Recognition and Annotation. In *ECCV*, 2014.

[12] M. Guillaumin and V. Ferrari. Large-scale Knowledge Transfer for Object Localization in ImageNet. In *CVPR*, 2012.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[14] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[15] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *T-PAMI*, 2013.

[16] J. Liu, B. Kuipers, and S. Savarese. Recognizing Human Actions by Attributes. In *CVPR*, 2011.

[17] M. Marszalek and C. Schmid. Semantic Hierarchies for Visual Object Recognition. In *CVPR*, 2007.

[18] D. Parikh and K. Grauman. Relative Attributes. In *ICCV*, 2011.

[19] M. Rohrbach, M. Stark, and B. Schiele. Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In *CVPR*, 2011.

[20] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. Learning to Share Visual Appearance for Multiclass Object Detection. In *CVPR*, 2010.

[21] B. Shahbaba and R. M. Neal. Improving Classification When a Class Hierarchy is Available Using a Hierarchy-Based Prior. *Bayesian Analysis*, 2:221–238, 2007.

[22] C. Wah and S. Belongie. Attribute-Based Detection of Unfamiliar Classes with Humans in the Loop. In *CVPR*, 2013.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[24] Y. Wang and G. Mori. A Discriminative Latent Model of Object Classes and Attributes. In *ECCV*, 2010.

[25] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-f. Chang. Designing Category-Level Attributes for Discriminative Visual Recognition . In *CVPR*, 2013.

[26] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV*, 2014.

[27] H. Zhang, Z.-j. Zha, Y. Yang, S. Yan, Y. Gao, and T.-S. Chua. Attribute-augmented Semantic Hierarchy. In *ACM MM*, 2013.

[28] A. Zweig and D. Weinshall. Exploiting Object Hierarchy: Combining Models from Different Category Levels. In *ICCV*, 2007.