Integrating Domain Knowledge: Using Hierarchies to Improve Deep Classifiers

Clemens-Alexander Brust[1][0000−0001−5419−1998] and Joachim Denzler[1,2]

[1] Computer Vision Group, Friedrich Schiller University Jena, Jena, Germany
[2] Michael Stifel Center Jena, Jena, Germany

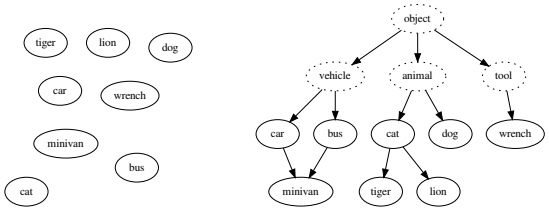
Abstract. One of the most prominent problems in machine learning in the age of deep learning is the availability of sufficiently large annotated datasets. For specific domains, *e.g.* animal species, a long-tail distribution means that some classes are observed and annotated insufficiently. Additional labels can be prohibitively expensive, *e.g.* because domain experts need to be involved. However, there is more information available that is to the best of our knowledge not exploited accordingly.
In this paper, we propose to make use of preexisting class hierarchies like WordNet to integrate additional domain knowledge into classification. We encode the properties of such a class hierarchy into a probabilistic model. From there, we derive a novel label encoding and a corresponding loss function. On the ImageNet and NABirds datasets our method offers a relative improvement of 10.4% and 9.6% in accuracy over the baseline respectively. After less than a third of training time, it is already able to match the baseline's fine-grained recognition performance. Both results show that our suggested method is efficient and effective.

Keywords: Class Hierarchy · Knowledge Integration · Hierarchical Classification

1 Introduction

In recent years, convolutional neural networks (CNNs) have achieved outstanding performance in a variety of machine learning tasks, especially in computer vision, such as image classification [15, 25] and semantic segmentation [27]. Training a CNN from scratch in an end-to-end fashion not only requires considerable computational resources and experience, but also large amounts of labeled training data [35]. Using pre-trained CNN features [33], adapting existing CNNs to new tasks [17] or performing data augmentation can reduce the need for labeled training data, but may not always be applicable or effective.

For specific problem domains, *e.g.* with a long-tailed distribution of samples over classes, the amount of labeled training data available is not always sufficient for training a CNN to reasonable performance. When unlabeled data already exists, which is not always the case, active learning [32] to select valuable instances for labeling may be applied. However, labels still have to be procured which is not always feasible.



(a) Class Set (b) Class Hierarchy

Fig. 1. Comparison between a loose set of independent classes and a class hierarchy detailing inter-class relations.

Besides information from more training data, domain knowledge in the form of high-level information about the structure of the problem can be considered. In contrast to annotations of training data, this kind of domain knowledge is already available in many cases from projects like iNaturalist [38], Visual Genome [23], Wikidata [41] and WikiSpecies[3].

In this paper, we use class hierarchies, *e.g.* WordNet [11], as an example of domain knowledge. In contrast to approaches based on attributes, where annotations are often expected to be per-image, class hierarchies offer the option of domain knowledge integration on the highest level with the least additional annotation effort. We encode the properties of such a class hierarchy into a probabilistic model that is based on common assumptions around hierarchies. From there, we derive a special label encoding together with a corresponding loss function. These components are applied to a CNN and evaluated systematically.

Our main **contributions** are: (i) a deep learning method based on a probabilistic model to improve existing classifiers by adding a class hierarchy which (ii) works with any form of hierarchy representable using a directed acyclic graph (DAG), *i.e.* does not require a tree hierarchy. We evaluate our method in experiments on the CIFAR-100 [24], ImageNet and NABirds [37] datasets to represent problem domains of various scales.

2 Related Work

Hierarchical methods have been subject of extensive research in image categorization. A given class hierarchy can be used explicitly to build a hierarchical classifier [20, 28], to regularize a preexisting model [12, 34], to construct an embedding space [2, 10, 13, 21], in metric learning-based methods [20, 40, 44] or, to construct a probabilistic model [7, 14].

Leveraging external semantic information for performance improvements, also called knowledge transfer, has been studied in the context of text categorization [3] as well as visual recognition [19, 30, 42]. Attributes are also considered

[3] `https://species.wikimedia.org/wiki/Main_Page`

as a knowledge source in [21]. While improvements are generally expected when using such methods, disagreements between visual and semantic similarity may introduce new errors [5]. Alternatively, visual hierarchies can be learned [1, 43] or used implicitly [4, 8]. An extreme case of knowledge transfer is zero-shot learning, where some categories have zero training examples [18, 31].

Our work is most closely related to [28] in that we consider similar individual classification problems. However, instead of their step-by-step approach using binary classifiers, our probabilistic model is evaluated globally for inference. A similar approach is also used in [7], where a relations between classes such as subsumption and mutual exclusion are extracted from a hierarchy and then used to condition a graphical model.

Hierarchical Data Typical image classification datasets rarely offer hierarchical information. There are exceptions such as the iNaturalist challenge dataset [38] where a class hierarchy is derived from biological taxonomy. Exceptions also include specific hierarchical classification benchmarks, *e.g.* [29, 36] as well as datasets where the labels originate from a hierarchy such as ImageNet [6]. The Visual Genome dataset [23] is another notable exception, with available metadata including attributes, relationships, visual question answers, bounding boxes and more, all mapped to elements from WordNet.

To augment existing non-hierarchical datasets, class hierarchies can be used. For a typical object classification scenario, concepts from the WordNet database [11] can be mapped to object classes. WordNet contains nouns, verbs and adjectives that are grouped into *synsets* of synonymous concepts. Relations such as hyponymy (**is-a**), antonymy (**is-not**), troponymy (**is-a-way-of**) and meronymy (**is-part-of**) are encoded in a graph structure where synsets are represented by nodes and relations by edges respectively. In this paper, we use the hyponymy relation to infer a class hierarchy.

3 Method

In this section, we propose a method to adapt existing classifiers to hierarchical classification. We start by acquiring a hierarchy and then define a probabilistic model based on it. From this probabilistic model, we derive an encoding and a loss function that can be used in a machine learning environment.

3.1 Class Hierarchy

For our model, we assume that a hierarchy of object categories is supplied, *e.g.* from a database such as WordNet [11] or WikiSpecies. It is modeled in the form of a graph $W = (S, h)$, where S denotes the set of all possible object categories, called *synsets* in the WordNet terminology. These are the nodes of the graph. Note that S is typically a superset of the dataset categories $C \subseteq S$, since parent categories are included to connect existing categories, *e.g.* `vehicle` is a parent of `car` and `bus`, but not originally part of the dataset.

A hyponymy relation $h \in S \times S$ over the classes, which can be interpreted as directed edges in the graph, is also given. For example, $(s, s') \in h$ means that s' is a hyperonym of s, or s is a hyponym of s', meaning s **is-a** s'. In general, the **is-a** relation is transitive. However, WordNet only models direct relationships between classes to keep the graph manageable and to represent different levels of abstraction as graph distances. The relation is also irreflexive and asymmetric.

For the following section, we assume that W is a directed acyclic graph (DAG). However, the WordNet graph is commonly reduced to a tree, for example by using a voting algorithm [36] or selecting task-specific subsets that are trees [6]. In this paper, we work on the directed acyclic graph (DAG) directly.

3.2 Probabilistic Model

Elements of a class hierarchy are not always mutually exclusive, *e.g.* a `corgi` is also a `dog` and an `animal` at the same time. Hence, we do not model the class label as one categorical random variable, but assume multiple independent Bernoulli variables $Y_s, s \in S$ instead. Formally, we model the probability of any class s occurring independently (and thus allowing even multi-label scenarios), given an example x:

$$P(Y_s = 1 | X = x), \tag{1}$$

or, more concisely,

$$P(Y_s^+ | X). \tag{2}$$

The aforementioned model on its own is overly flexible considering the problem at hand, since it allows for any combination of categories co-occurring. At this point, assumptions are similar to those behind a one-hot encoding. However, from the common definition of a hierarchy, we can infer a few additional properties to restrict the model.

Hierarchical decomposition A class s can have many independent parents $S' = s'_1, \ldots, s'_n$. We choose $Y_{S'}^+$ to denote an observation of at least one parent and $Y_{S'}^-$ to indicate that no parent class has been observed:

$$\begin{aligned} Y_{S'}^+ &\Leftrightarrow Y_{s'_1}^+ \vee \ldots \vee Y_{s'_n}^+ \Leftrightarrow Y_{s'_1} = 1 \vee \ldots \vee Y_{s'_n} = 1, \\ Y_{S'}^- &\Leftrightarrow Y_{s'_1}^- \wedge \ldots \wedge Y_{s'_n}^- \Leftrightarrow Y_{s'_1} = 0 \wedge \ldots \wedge Y_{s'_n} = 0. \end{aligned}$$

Based on observations $Y_{S'}$, we can decompose the model from Equation (2) in a way to capture the hierarchical nature. We start by assuming a marginalization of the conditional part of the model over the parents $Y_{s'}$:

$$\begin{aligned} P(Y_s^+ | X) = {} & P(Y_s^+ | X, Y_{S'}^+) P(Y_{S'}^+ | X) \\ & + P(Y_s^+ | X, Y_{S'}^-) P(Y_{S'}^- | X). \end{aligned} \tag{3}$$

The details of this decomposition are given in the supplementary material.

Simplification We now constrain the model and add assumptions to better reflect the hierarchical problem. If none of the parents $S' = s'_1, \ldots, s'_n$ of a class s occur, we assume the probability of s being observed for any given example to be zero:

$$P(Y_s^+|X, Y_{S'}^-) = P(Y_s^+|Y_{S'}^-) = 0. \tag{4}$$

This leads to a simpler hierarchical model, omitting the second half of Equation (3) by setting it to zero:

$$P(Y_s^+|X) = P(Y_s^+|X, Y_{S'}^+)P(Y_{S'}^+|X). \tag{5}$$

Parental independence To make use of recursion in our model, we require the random variables $Y_{s'_1}, \ldots, Y_{s'_n}$ to be independent of each other in a naive fashion. Using the definition of $Y_{S'}^+$, we derive:

$$P(Y_{S'}^+|X) = 1 - \prod_{i=1}^{|S'|} 1 - P(Y_{s'_i}^+|X). \tag{6}$$

Parentlessness In a non-empty DAG, we can expect there to be at least one node with no incoming edges, *i.e.* a class with no parents. In the case of WordNet, there is exactly one node with no parents, the root synset `entity.n.01`. A marginalization over parent classes does not apply there. We assume that all observed classes are children of `entity` and thus set the probability to one for a class without parents:

$$P(Y_s^+|X, S' = \emptyset) = 1. \tag{7}$$

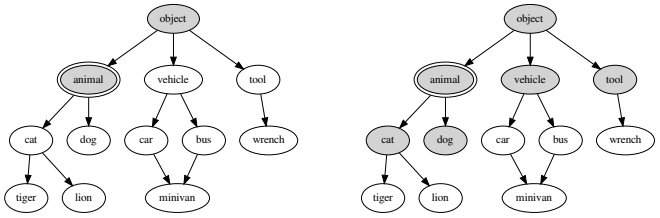
Note that this is not reasonable for all hierarchical classification problems. If the hierarchy is composed of many disjoint components, $P(Y_s^+|X, S' = \emptyset)$ should be modeled explicitly. Even if there is only a single root, explicit modeling could be used for tasks such as novelty detection.

3.3 Inference

The following section describes the details of the inference process in our model.

Restricted Model Outputs Depending on the setting, when the model is used for inference, the possible outputs can be restricted to the classes C that can actually occur in the dataset as opposed to all modeled classes S including parents that exist only in the hierarchy. This assumes a known class set at test time as opposed to an open-set problem. We denote this setting *mandatory labeled node prediction (MLNP)* and the unrestricted alternative *arbitrary node prediction (ANP)*.

Prediction To predict a *single* class s given a specific example x, we look for the class where the joint probability of the following observations is high: (i) the class s itself occurring (Y_s^+) and (ii) none of the children $S'' = s''_1, \ldots, s''_m$ occurring ($Y_{S''}^-$):



(a) Encoding $e(y)$ (b) Loss Mask $m(y)$

Fig. 2. Hierarchical encoding and loss mask for $y = \texttt{animal}$. Shaded nodes represent 1 and light nodes 0 respectively.

$$s(x) = \underset{s \in C \subseteq S}{\operatorname{argmax}} P(Y_s^+|X) P(Y_{S''}^-|X, Y_s^+). \tag{8}$$

Requiring the children to be pairwise independent similar to Equation (6), inference is performed in the following way:

$$s(x) = \underset{s \in C \subseteq S}{\operatorname{argmax}} P(Y_s^+|X) \prod_{i=1}^{|S''|} 1 - P(Y_{s_i''}^+|X, Y_s^+). \tag{9}$$

Because $P(Y_s^+|X)$ can be decomposed according to Equation (3) and expressed as a product (cf. Equation (6)), we infer using:

$$s(x) = \underset{s \in C \subseteq S}{\operatorname{argmax}} P(Y_s^+|X, Y_{S'}^+) \cdot \underbrace{(1 - \prod_{i=1}^{|S'|} 1 - P(Y_{s_i'}^+|X))}_{\text{Parent nodes } S'} \cdot \underbrace{\prod_{i=1}^{|S''|} 1 - P(Y_{s_i''}^+|X, Y_s^+)}_{\text{Child nodes } 'S}. \tag{10}$$

Again, $P(Y_{s_i'}^+|X)$ can be decomposed. This decomposition is performed recursively following the scheme laid out in Equation (3) until a parentless node is reached (cf. Equation (7)).

3.4 Training

In this section, we describe how to implement our proposed model in a machine learning context. Instead of modeling the probabilities $P(Y_s^+|X)$ for each class s directly, we want to estimate the conditional probabilities $P(Y_s^+|X, Y_{S'}^+)$. This changes each individual estimator's task slightly, because it only needs to discriminate among siblings and not all classes. It also enables the implementation of the hierarchical recursive inference used in Equation (10).

The main components comprise of a label encoding $e : S \to \{0,1\}^{|S|}$ as well as a special loss function. A label $y \in S$ is encoded using the hyponymy relation $h \in S \times S$, specifically its transitive closure $\mathcal{T}(h)$, and the following function:

$$e(y)_s = \begin{cases} 1 & \text{if } y = s \text{ or } (y,s) \in \mathcal{T}(h), \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

A machine learning method can now be used to estimate encoded labels directly. However, a suitable loss function needs to be provided such that the conditional nature of each individual estimator is preserved. This means that, given a label y, a component s should be trained only if one of its parents s' is related to the label y by $\mathcal{T}(h)$, or if y is one of its parents. We encode this requirement using a *loss mask* $m : S \to \{0,1\}^{|S|}$, defined by the following equation:

$$m(y)_s = \begin{cases} 1 & y = s \text{ or} \\ & \exists (s,s') \in h : y = s' \text{ or } (y,s') \in \mathcal{T}(h), \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Figure 2 visualizes the encoding $e(y)$ and the corresponding loss mask $m(y)$ for a small example hierarchy. Using the encoding and loss mask, the complete loss function $\mathcal{L}$ for a given data point (x,y) and estimator $f : X \to \{0,1\}^{|S|}$ is then defined by the the masked mean squared error (alternatively, a binary cross-entry loss can be used):

$$\mathcal{L}_f(x,y) = m(y)^T (e(y) - f(x))^2. \tag{13}$$

The function $f(x)_s$ is then used to estimate the conditional probabilities $P(Y_s^+ | X, Y_{S'}^+)$. Applying the inference procedure in Section 3.3, a prediction is made using the formula in Equation (10) and substituting $f(x)_s$ for $P(Y_s^+ | X, Y_{S'}^+)$.

4 Experiments

We aim to assess the effects of applying our method on three different scales of problems, using the following datasets:

CIFAR-100 For our experiments, we want to work with a dataset that does not directly supply hierarchical labels, but where we can reasonably assume that an underlying hierarchy exists. The CIFAR-100 dataset [24] fulfills this requirement. Because there are only 100 classes, each can be mapped to a specific synset in the WordNet hierarchy without relying on potentially faulty automation. Direct mapping is not always possible, *e.g.* `aquarium_fish`, which doesn't exist in WordNet and was mapped to `freshwater fish.n.01` by us. This process is a potential error source.

The target hierarchy is composed in three steps. First, the synsets mapped from all CIFAR-100 classes make up the foundation. Then, parents of the synsets are added in a recursive fashion. With the nodes of the graph complete, directed edges are determined using the WordNet hyponymy relation. Mapping all classes to the WordNet synsets results in 99 classes being mapped to leaf nodes and one class to an inner node (`whale`). In total, there are 267 nodes as a result of the recursive adding of hyperonyms.

ImageNet Manually mapping dataset labels to WordNet synsets is a potentional source of errors. An ideal dataset would use WordNet as its label space. Because of WordNet's popularity, such datasets exist, *e.g.* ImageNet [6] and 80 Million Tiny Images [36]. We use ImageNet, specifically the dataset of the 2012 ImageNet Large Scale Visual Recognition Challenge. It contains around 1000 training images each for 1000 synsets. All 1000 synsets are leaf nodes in the resulting hierarchy with a total of 1860 nodes.

NABirds Quantifying performance on object recognition datasets such as CIFAR and ImageNet is important to prove the general usefulness of a method. However, more niche applications such as fine-grained recognition stand to benefit more from improvements because the availability of labeled data is much more limited. We use the NABirds dataset [37] to verify our method in a fine-grained recognition setting. NABirds is a challenge where 555 categories of North American birds have to be differentiated. These categories are comprised of 404 species as well as several variants of sex, age and plumage. It contains 48,562 images split evenly into training and validation sets. Annotations include not only image labels, but also bounding boxes and parts. Additionally, a class hierarchy based on taxonomy is supplied. It contains 1010 nodes, where all of the 555 visual categories are leaf nodes.

4.1 Experimental Setup

Convolutional Neural Networks For our experiments on the CIFAR-100 dataset, we use a ResNet-32 [15] in the configuration originally designed for CIFAR. The network is initialized randomly as specified in [15].

We use a minibatch size of 128 and the adaptive stochastic optimizer Adam [22] with a constant learning rate of 0.001 as recommended by the authors. Although SGD can lead to better performance of the final models, its learning rate is more dependent on the range of the loss function. We choose an adaptive optimizer to minimize the influence of different ranges of loss values.

In our NABirds and ImageNet experiments, we use a ResNet-50 [15, 16] because of the larger image size and overall scale of the dataset. The minibatch size is reduced to 64 and training is extended to $120,000$ steps for NABirds and $234,375$ steps for ImageNet. We crop all NABirds images using the given bounding box annotations and resize them to 224×224 pixels.

All settings use random shifts of up to 4 pixels for CIFAR-100 and up to 32 pixels for NABirds and ImageNet as well as random horizontal flips during training. All images are normalized per channel to zero mean and standard deviation

one, using parameters estimated over the training data. Code will be published along with the paper. We choose our ResNet-50 and ResNet-32 baselines to be able to judge effects across datasets, which would not be possible when selecting a state-of-the-art method for each dataset individually. Furthermore, the moderately sized architecture enables faster training and therefore more experimental runs compared to a high performing design such as PNASNet [26].

Evaluation We report the overall accuracy, not normalized w.r.t class instance counts. Each experiment consists of six random initializations per method for the CIFAR-100 dataset and three for the larger-scale NABirds and ImageNet datasets, over which we report the average. We choose to compare the methods using a measure that does not take hierarchy into account to gauge the effects of adding hierarchical data to a task that is not normally evaluated with a specific hierarchy in mind. Using a hierarchical measure would achieve the opposite: we would measure the loss sustained by omitting hierarchical data.

4.2 Overall Improvement — ImageNet

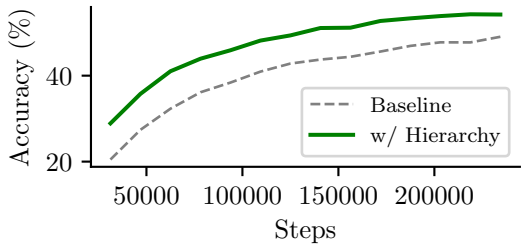


Fig. 3. Accuracy on the ImageNet validation set over time. Our hierarchical training method gains accuracy faster than the flat classifier baseline. We report overall classification accuracy in percent.

In this experiment, we quantify the effects of using our hierarchical classification method to replace the common combination of one-hot encoding and mean squared error loss function. We use ImageNet, specifically the ILSVRC2012 dataset. This is a classification challenge with 1000 classes whose labels are taken directly from the WordNet hierarchy of nouns.

Figure 3 shows the evolution over time of accuracy on the validation set. After around 240,000 gradient steps, training converges. The one-hot baseline reaches a final accuracy of 49.1%, while our method achieves 54.2% with no changes to training except for our loss function and hierarchical encoding. This is a relative improvement of 10.4%.

While an improvement of accuracy at the end of training is always welcome, the effects of hierarchical classification more drastically show in the change in

accuracy over time. The strongest improvement is observed during the first training steps. After training for 31250 steps using our method, the network already performs with 28.9% accuracy. The one-hot baseline matches this performance after 62500 gradient steps, taking twice as long. The baseline's final accuracy of 49.1% is matched by our method after only $125,000$ training steps, resulting in an overall training speedup of 1.88x.

4.3 Speedup — CIFAR-100

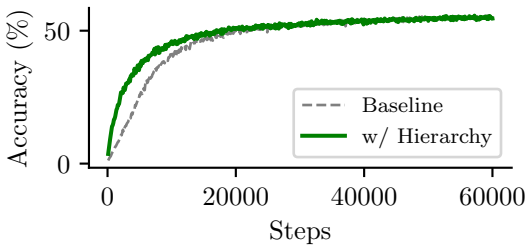


Fig. 4. Results on the CIFAR-100 validation set. Our hierarchical training method gains accuracy faster than the flat classifier baseline. We report overall classification accuracy in percent.

We report the accuracies on the validation set as they develop during training in Figure 4. As training converges, we observe almost no difference between both methods, with our hierarchical method reaching 54.6% and the one-hot encoding baseline at 55.4%. However, the methods differ strongly in the way that accuracy is achieved. After the first 500 steps, our hierarchical classifier already predicts 10.7% of the validation set correctly, compared to the baseline's 2.8%. It takes the baseline another 1600 steps to match 10.7%, or 4.2 times as many steps.

This advantage in training speed is very strong during initial training, but becomes smaller over time. After the first half of training, the difference between both methods vanishes almost completely.

4.4 Fine-Grained Recognition — NABirds

To evaluate the performance of our hierarchical method in a more specific setting, we use the NABirds dataset [37], a fine-grained recognition challenge where the task is to classify 555 visual categories of birds. A hierarchy is given by the dataset. We observe results similar to the ImageNet dataset (see Section 4.2), where our method leads to an improvement in both training speed and overall accuracy. The one-hot baseline converges to an accuracy of 56.5%. Our hierarchical classifier reaches 61.9% after the full $120,000$ steps of training. It already matches the baseline's final accuracy at $39,000$ iterations, reducing training time to less than a third. The relative improvement with full training is 9.6%.

Table 1. Results Overview.

Dataset	# of classes	Accuracy (%) Baseline	Accuracy (%) w/Hierarchy	Speedup w/Hierarchy Overall	Speedup w/Hierarchy Initial
CIFAR-100	100	**55.4** $\pm$ 0.84	54.6 $\pm$ 1.03	—	7.00
NABirds	555	56.5 $\pm$ 0.49	**61.9** $\pm$ 0.27	3.08	10.00
ILSVRC2012	1000	49.1 $\pm$ 0.33	**54.2** $\pm$ 0.04	1.88	—

4.5 Overview and Discussion

Table 1 provides the most important facts for each dataset. We report the accuracy at the end of training for the one-hot baseline as well as our method. Overall speedup indicates how much faster in terms of training steps our hierarchical method achieves the end-of-training accuracy of the baseline. Initial speedup looks at the accuracy delivered by our method after the first validation interval. We then measure how much longer the baseline needs to catch up.

On all 3 datasets, the initial training is faster using our method. However, we only observe an improvement in classification accuracy on ImageNet and NABirds. With CIFAR-100, the benefits of adding hierarchical information are limited to training speed. There are a few possible explanations for this:

First, the CIFAR-100 dataset is the only dataset that requires a manual mapping to an external hierarchy, whereas the other datasets either supply one or have labels directly derived from one. The manual mapping is a possible error source and as such, could explain the observation, as could the small image size.

The second possible reason lies in the difference between semantic similarity and visual similarity [5, 9]. Semantic similarity relates two classes using their meaning. It can be extracted from hierarchies such as WordNet [11], for example by looking at distances in the graph. Visual similarity on the other hand relates images that look alike, regardless of the meaning behind them. When classifying, we group images by semantics, even if they share no visual characteristics. Adding information based on only semantics can thus lead to problems.

5 Conclusion

We present a method to modify existing deep classifiers such that knowledge about relationships between classes can be integrated. The method is derived from a probabilistic model that is itself based on our understanding of the meaning of hierarchies. Overall, it is just one example of the integration of domain knowledge in an otherwise general method. One could also consider our method a special case of learning using privileged information [39].

Our method can improve classifiers by utilizing information that is freely available in many cases such as WordNet [11] or WikiSpecies. There are also datasets which include a hierarchy that is ready to use [6, 37].

Further research should focus on the data insufficiency aspect and quantify the data reduction made possible by our method on small datasets, and compare the sample efficiency to the baseline for artificially reduced datasets as well as alternatives such as data augmentation.

References

[1] E. Bart et al. "Unsupervised learning of visual taxonomies". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8.

[2] B. Barz and J. Denzler. "Hierarchy-Based Image Embeddings for Semantic Image Retrieval". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Jan. 2019, pp. 638–647.

[3] Mohammed Benkhalifa, Abdelhak Mouradi, and Houssaine Bouyakhf. "Integrating WordNet knowledge to supplement training data in semi-supervised agglomerative hierarchical clustering for text categorization". In: *International Journal of Intelligent Systems* 16.8 (2001), pp. 929–947.

[4] A. Bilal et al. "Do Convolutional Neural Networks Learn Class Hierarchy?" In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 152–162.

[5] Clemens-Alexander Brust and Joachim Denzler. "Not just a matter of semantics: the relationship between visual similarity and semantic similarity". In: *arXiv:1811.07120 [cs]* (Nov. 17, 2018). arXiv: 1811.07120.

[6] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255.

[7] Jia Deng et al. "Large-Scale Object Classification Using Label Relation Graphs". In: *European Conference on Computer Vision (ECCV)*. Vol. 8689. Cham: Springer International Publishing, 2014, pp. 48–64.

[8] Jia Deng et al. "What Does Classifying More Than 10,000 Image Categories Tell Us?" In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 71–84.

[9] Thomas Deselaers and Vittorio Ferrari. "Visual and semantic similarity in imagenet". In: *Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 1777–1784.

[10] Fartash Faghri et al. "VSE++: Improving Visual-Semantic Embeddings with Hard Negatives". In: *arXiv:1707.05612 [cs]* (July 18, 2017). arXiv: 1707.05612.

[11] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

[12] Rob Fergus et al. "Semantic Label Sharing for Learning with Many Categories". In: *Computer Vision – ECCV 2010*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 762–775.

[13] Andrea Frome et al. "DeViSE: A Deep Visual-Semantic Embedding Model". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 2121–2129.

[14] Eric Gaussier et al. "A hierarchical model for clustering and categorising documents". In: *European Conference on Information Retrieval*. Springer, 2002, pp. 229–247.

[15] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Computer Vision and Pattern Recognition (CVPR)*. 2016.

[16] Kaiming He et al. "Identity mappings in deep residual networks". In: *European Conference on Computer Vision (ECCV)*. 2016, pp. 630–645.

[17] Judy Hoffman et al. "LSDA: Large Scale Detection Through Adaptation". In: *arXiv preprint arXiv:1407.5035*. July 18, 2014. arXiv: 1407.5035v3.

[18] Yuqi Huo et al. "Zero-Shot Learning with Superclasses". In: *Neural Information Processing*. Ed. by Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa. Lecture Notes in Computer Science. Springer International Publishing, 2018, pp. 460–472.

[19] Sung Ju Hwang. "Discriminative object categorization with external semantic knowledge". PhD thesis. Aug. 2013.

[20] Sung Ju Hwang, Kristen Grauman, and Fei Sha. "Learning a Tree of Metrics with Disjoint Visual Features". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 621–629.

[21] Sung Ju Hwang and Leonid Sigal. "A Unified Semantic Embedding: Relating Taxonomies and Attributes". In: *Advances in Neural Information Processing Systems 27*. 2014, p. 9.

[22] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference for Learning Representations (ICLR)*. Dec. 22, 2014. arXiv: 1412.6980v9.

[23] Ranjay Krishna et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International Journal of Computer Vision (IJCV)* 123.1 (2017), pp. 32–73.

[24] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1097–1105.

[26] Chenxi Liu et al. "Progressive neural architecture search". In: *arXiv preprint arXiv:1712.00559*. 2017.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015. arXiv: 1411.4038v2.

[28] M. Marszalek and C. Schmid. "Semantic Hierarchies for Visual Object Recognition". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. June 2007, pp. 1–7.

[29] Ioannis Partalas et al. "LSHTC: A benchmark for large-scale text classification". In: *arXiv preprint arXiv:1503.08581* (2015).

[30] Erik Rodner and Joachim Denzler. "One-Shot Learning of Object Categories Using Dependent Gaussian Processes". In: *Pattern Recognition*. Ed. by Michael Goesele et al. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 232–241.

[31] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. "Transfer Learning in a Transductive Setting". In: *Advances in Neural Information Processing*

Systems 26. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 46–54.

[32] Burr Settles. *Active Learning Literature Survey*. Tech. rep. 1648. University of Wisconsin–Madison, 2009.

[33] Ali Sharif Razavian et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition". In: *Computer Vision and Pattern Recognition Workshops (CVPR-WS)*. 2014.

[34] Nitish Srivastava and Ruslan R Salakhutdinov. "Discriminative Transfer Learning with Tree-based Priors". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 2094–2102.

[35] Chen Sun et al. "Revisiting unreasonable effectiveness of data in deep learning era". In: *International Conference on Computer Vision (ICCV)*. 2017, pp. 843–852.

[36] Antonio Torralba, Rob Fergus, and William T Freeman. "80 million tiny images: A large data set for nonparametric object and scene recognition". In: *Transactions on Pattern Analysis and Machine Intelligence (PAMI)* 30.11 (2008), pp. 1958–1970.

[37] Grant Van Horn et al. "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 595–604.

[38] Grant Van Horn et al. "The iNaturalist Challenge 2017 Dataset". In: *arXiv preprint arXiv:1707.06642*. 2017.

[39] Vladimir Vapnik and Akshay Vashist. "A new learning paradigm: Learning using privileged information". In: *Neural Networks* 22.5-6 (2009), pp. 544–557.

[40] N. Verma et al. "Learning hierarchical similarity metrics". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. June 2012, pp. 2280–2287.

[41] Denny Vrandečić and Markus Krötzsch. "Wikidata: a free collaborative knowledgebase". In: *Communications of the ACM* 57.10 (2014), pp. 78–85.

[42] Q. Wu et al. "Image Captioning and Visual Question Answering Based on Attributes and External Knowledge". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (June 2018), pp. 1367–1381.

[43] Zhicheng Yan et al. "HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 2740–2748.

[44] Xiaofan Zhang et al. "Embedding Label Structures for Fine-Grained Feature Representation". In: 2016, pp. 1114–1123.