Improving Image Understanding with Concept Relation Graph

Libo Yin

A subthesis submitted in partial fulfillment of the degree of
Bachelor of Information Technology (Honours) at
The Department of Computer Science
Australian National University

October 2015

Typeset in Palatino by $\TeX$ and $\LaTeX\,2_{\varepsilon}$.

Except where otherwise indicated, this thesis is my own original work.

Libo Yin
22 October 2015

Acknowledgements

This work is dedicated to:	Dr. Scott Sanner
	for supervising my first research project in 2013/14
	Miss. Qianyu Zhang
	for being an important step in my personal Odyssey
With special thanks to:	Dr. Lin Yin and Mrs. Yichun Zheng
	my parents
	Dr. Lexing Xie and Dr. Xuming He (NICTA)
	my supervisors
	Bruce Hall and Residents
	for being my home and family in the past 1.5 years

Abstract

This work considers image classification problem when there is semantic structure in the concept space. An effective framework to solve such problem, the HEX model, has been discussed in a prior work [4]. The HEX framework models the hierarchical, exclusive, and independence relationship within the extended concept space of dataset with a graphical model. By interpreting the semantic graphical model as a conditional random field (CRF), efficient structured inference algorithms have been developed. This work identifies the weakness of the original HEX model under the realistic labelling assumption, when a high percentage of images are labelled not to specific concepts, but to more general ones. Improvements of this work are delivered in multiple stages, by gradually allowing more and stronger relationships in the HEX model. Experimental results show significant improvements over the baseline and the original HEX model.

Contents

Chapter 1

Introduction

Image classification is a fundamental problem in computer vision and machine learning. It forms the building block of other algorithms such as object recognition and localisation. As a benchmark, the first subtask (three in total) of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [19] states:

> Classification: For each image, an algorithm will produce a list of at most 5 object categories in the descending order of confidence.

On the other hand, the first subtask (two in total) of the PASCAL Visual Object Classes Challenge (PASCAL VOC) 2012 [7] states:

> Classification: For each of the twenty classes, predicting presence/absence of an example of that class in the test image.

The comparison between the two aforementioned challenges reveal the two branches of multiclass classification: exclusive, where the classifier predicts exactly one out of all possible states to be true; and independent, where each state is predicted true or false independently. Both of them are considered “flat” classification, which means there is little structure within the concept space.

However, a completely flat concept space is highly unlikely. Even if there is no explicit structure in the concept space[1], some concepts are still more tightly connected than others. For example, consider concept space {Persian, British Shorthair, Labrador, Husky}. The former two are cats, and the latter two are dogs. Concepts sharing a common hypernym, i.e. within the same group, share more visual characteristics than otherwise. Meanwhile, these four concepts are also mutually exclusive. For a large dataset such as ImageNet, there exists rich structure within the concept space.

This issue calls for a classification framework that captures more relationships between concepts. A recent advancement in this direction is [4]. It models the hierarchical, exclusive, and independent relationship between concepts with a semantic graphical model. In order to capture implicit relationships among concepts that share

[1] A classic example of concept space with explicit structure is {dog, puppy, Husky, cat}. In this case, puppy and Husky are independent subcategories of dog; and an object cannot be both a cat and a dog. Neither exclusive nor independent multiclass classification is able to capture such structure.

a common hypernym, the concept space is extended by the hypernyms of concepts. This framework allows an image to be classified to a semantically consistent hierarchy, hence reaching a balance between the overly relaxed independent multiclass classification, and the overly conservative exclusive multiclass classification. Finally, the graphical model can be interpreted as a CRF, allowing efficient structured inference algorithms to be developed. For details of the inference algorithm, refer to chapter 5.3.

This work is built on top of [4]. While a theoretically sound framework has been laid, this work focuses on the weakness of the original HEX model under severe dataset imbalance: The original system cannot cope with the situation where a high percentage of images are labelled not to specific concepts, but to more general ones. Having identified the problem, this work proposed enhancements to the system by gradually encoding more and stronger relationships to the potential function of CRF, and by allowing parameter learning on the CRF. The improvements are empirically tested on a modified version of the PASCAL dataset. Thanks to the small size of PASCAL (compared to ImageNet), this work is able to present some detailed observations with regard to the dynamics of the HEX framework.

The structure of this document is as follows: Chapter 2 contains a short review of previous publications related to this work. Chapter 3 introduces the original HEX model, as well as its weaknesses. Chapter 4 covers the reimplementation details for this work, such as the dataset setup. Finally, the improvements of this work is derived, explained, and empirically tested in Chapter 5.

Chapter 2

Related Works

The baseline algorithm to solve image classification problem is the convolutional neural network (CNN). After a major breakthrough in 2012 [11], CNN has gained higher performance by adding more layers [21]. New training methods have been developed to accommodate increasingly complex CNN [20]. The development of DeConvNet [24] allowed each layer of CNN to be visualised, greatly benefiting the tuning of network structure. In addition to solving image classification problem directly, CNN has been proven to be an effective attribute extractor in [17]. In [4, 5], this feature was applied to zero-shot classification [6].

Exploiting structures in the concept space is not a new idea [22]. Modelling hierarchical relationship between concepts has been explored in [13, 15]; and modelling exclusive relationships has been attempted in [2]. In 2014, [4] became the first work to have jointly modelled the hierarchical and exclusive relationship between concepts with automatically generated relation graph. [5] extended [4] to allow non-deterministic relationship between concepts. Similar idea has been attempted in [23], which modelled the probabilistic cooccurrence and exclusive relationship among concepts with a manually created semantic Bayesian network. With semantic relationship modelled, the structured prediction problem is usually solved with CRF [12] or structured SVM.

The modelling of entity relationships is not limited to semantic relationship among concepts. Similar techniques have a longer history on object recognition tasks. For example, combining the cooccurrence relationship and the locational relationship of objects in a scene has been attempted in [3, 8]. Finally, part of this work falls in the category of transfer learning [18].

Chapter 3

The Original HEX Model

3.1 Structure

The original HEX model [4] is an extension of the baseline flat multiclass classification framework. It models the Hierarchical, EXclusive, and independent relationship within the concept space of dataset, extended by their hypernyms (in order to capture implicit relationships among concepts that belong to the same "group", as discussed in chapter 1), with a semantic graphical model. Each node in the HEX graph corresponds to a concept in the extended concept space being active (true) or inactive (false). States of neighbouring nodes are constrained in that if a is a hypernym of b, then is not allowed that $a = 0,\ b = 1$; and if a and b are exclusive, then they cannot both be true. The HEX framework classifies an image into a joint assignment of the state of nodes, corresponding to a hierarchy in the graphical model, that satisfies the above semantic consistency. A simple HEX graph is shown in figure 3.1.

Denoting the set of vertices by V, the set of hierarchical edges by E_h, and the set of exclusive edges by E_e, the joint assignment $y \in \{0,1\}^V$ can be defined as a CRF:

$$\tilde{p}(y|x) = \prod_{i \in V} \exp\{x_i \cdot I[y_i = 1]\} I[y \text{ legal}] \tag{3.1}$$

where unary input x_i is the confidence on concept i, provided by an arbitrary underlying classifier[1]. Local semantic constraints on hierarchical and exclusive edges is formalised by:

$$I[y \text{ legal}] = \prod_{(i,j) \in E_h} I[(y_i, y_j) \neq (0,1)] \prod_{(i,j) \in E_e} I[(y_i, y_j) \neq (1,1)] \tag{3.2}$$

Thanks to these semantic constraints, the state space of a HEX graph, i.e. the set of all legal joint assignments, is significantly smaller than the state space of independent multiclass classification on the same concept space. For example, with local confidence given beside each node in figure 3.1, the legal state space of the HEX graph and their respective unnormalised potentials are shown in table 3.1.

[1] According to [4]. It shall be explained in chapter 3.2 that the choice of underlying unary classifier is actually not arbitrary.

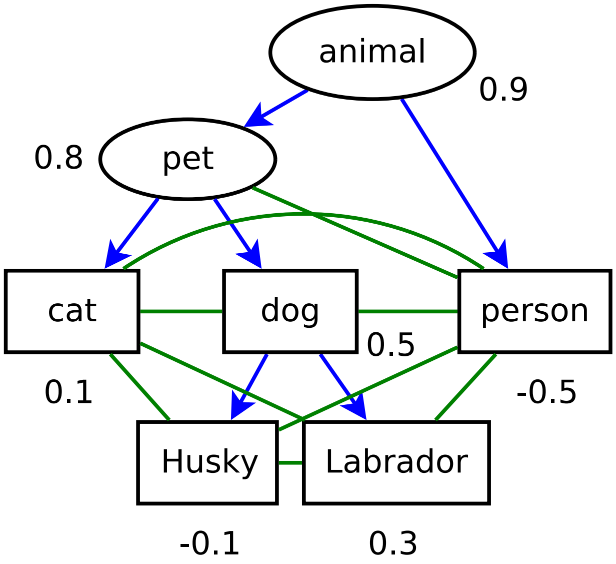

Figure 3.1: A simple HEX graph with 5 nodes in the original concept space (in rectangles) and 2 of their hypernyms (in ellipsoids). Directed edges denote semantical subsumption: $(a \to b)$ if a is a hypernym of b according to WordNet [14]; and undirected edges denote exclusion. Since exclusive relationship is not covered by WordNet, the exclusive subgraph is initialized greedily: two concepts are exclusive unless they share a common descendant in the hierarchical subgraph. Note that the hierarchical subgraph is in general a DAG rather than a tree; and it is not necessarily the case that all concepts in the original concept space are bottom-level nodes in the hierarchical subgraph. Values beside each nodes are to be used in table 3.1.

assignment	$\tilde{p}(y\|x)$
$\varnothing$	$\exp(0)$
{animal}	$\exp(0.9)$
{animal, pet}	$\exp(0.9+0.8)$
{animal, person}	$\exp(0.9-0.5)$
{animal, pet, cat}	$\exp(0.9+0.8+0.1)$
{animal, pet, dog}	$\exp(0.9+0.8+0.5)$
{animal, pet, dog, Husky}	$\exp(0.9+0.8+0.5-0.1)$
{animal, pet, dog, Labrador}	$\exp(0.9+0.8+0.5+0.3)$

Table 3.1: The extended state space, i.e. all valid assignments of the HEX graph in figure 3.1, and their respective potentials. In this example, the most likely joint assignment is {animal, pet, dog, Labrador}. The softmax baseline classifier, in the language of HEX, classifies to the original state space: {animal, pet, cat}, {animal, pet, dog}, {animal, pet, dog, Husky}, {animal, pet, dog, Labrador}, and {animal, person}.

Finally, an important assumption of [4] is that mechanical Turks tend to label an image to more abstract concepts. For example, an image of a yellow Labrador is more likely to be labelled to "dog" than "Labrador". Such realistic labelling behaviour is modelled by randomly relabelling images to their immediate parents. A major task of the HEX framework is to make accurate predictions when the relabelling rate is high.

3.2 Observations

Note that a valid assignment does not have to have an active node in the original concept space. (Actually, it does not have to have an active node at all, as $\varnothing$ is a valid joint assignment according to (3.2).) This allows an image to be classified to a joint assignment in which all active concepts are abstract. For example, in figure 3.1, to $\{\varnothing\}$, {animal}, or {animal, pet}. This is by no doubt a desirable feature for deployment. However, since all images are labelled in the original concept space, classifying to the extended concept space makes performance evaluation troublesome. While this issue is not addressed in [4], the evaluation strategy used for this work will be discussed in chapter 4.3.

Also note that in table 3.1, the competition between assignment {animal, pet, dog, Husky} and {animal, pet, dog, Labrador} depends entirely on the confidence of node "Husky" and "Labrador". However, to discriminate between {animal, pet, dog, Husky} and {animal, person} requires examining the confidence along different paths. From this example it is clear that, during the testing stage, confidence is passed down the hierarchy from more abstract concepts to more concrete ones. The other side of the same coin is that, during the training stage of the underlying unary classifier, a node corresponding to an abstract concept receives all training data of its children. This can be seen as confidence being passed up in the hierarchy. Such bidirectional propagation of confidence explains the advantage of the HEX framework under the realistic labelling assumption. Note that the exclusive subgraph does not take part in the propagation of confidence.

The third observation is a combined consequence of the greedy exclusion setup, as discussed under figure 3.1, and the greedy nature of the potential function (3.1). The prerequisite for the original HEX model is that the decision boundary of x_i is zero for all i: A unary prediction above zero gives support to a node being active, whereas a unary prediction below zero gives support to that node being inactive. In [4] this is not a problem, as a CNN [11] is used as the underlying unary classifier; and a neural network can learn its decision boundary from training data. However, if a probabilistic classifier is used, in which case $x_i \in [0,1]$ and the decision boudary is 0.5, then it is guaranteed that a bottom-level concept will be activated (see appendix A.1 for proof). This means that the effective state space size of the HEX graph is reduced to the number of bottom-level nodes, which is no larger than the original state space size. This defeats the purpose of HEX, although a smaller state space means faster inference and higher accuracy.

Finally, listed below are two less important observations:

1. There are no learnable variables in this CRF. In other words, all learning is performed in the underlying unary classifier.
2. Mathematically, CRF requires $\forall y : \tilde{p}(y|x) > 0$. However, computationally, assigning zero to $\tilde{p}(y|x)$ can be interpreted as assigning an infinitesimal value. Therefore, the above definition is computationally a legitimate CRF.

3.3 Problems

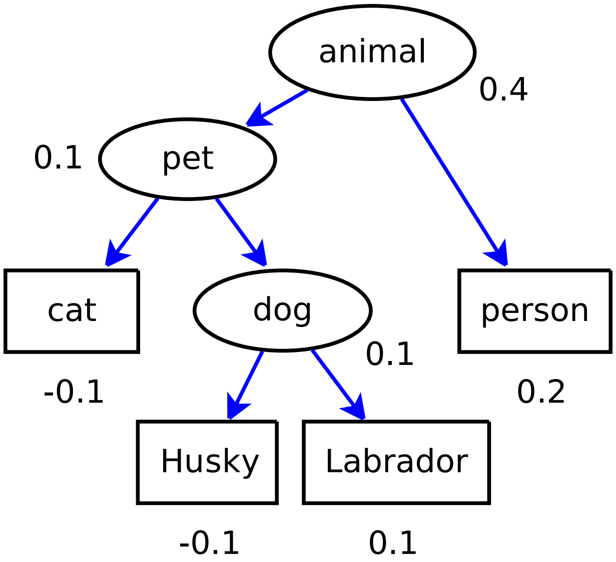

Figure 3.2: Where an assignment with more active nodes of lower confidence wins over an assignment with fewer active nodes of higher confidence. Exclusive edges are not drawn, as they do not carry further information once the state space has been calculated.

Consider the situation illustrated in figure 3.2. Following the same logic as table 3.1, the original HEX model predicts {animal, pet, dog, Labrador}. However, this is a result of more active nodes rather than active nodes of higher confidence. Intuitively, {animal, person} seems a more reasonable prediction. In addition, the two aforementioned assignments are only separated by 0.1 in log unnormalised potential. It will be desirable if the model can make a prediction that is further from the decision boundary by taking into account the confidence of other nodes, especially those with unary predictions far from the decision boundary.

The second problem is a combined consequence of the realistic labelling assumption and using an independent multiclass classifier as the underlying unary classifier. When the relabelling rate is high, most images are relabelled to their immediate parents. This means that for each concept X, most instances of X are correctly labelled to all its ancestors, but incorrectly labelled "NOT X". This create a highly unbalanced and internally inconsistent dataset. In such case, for images of X, the unary prediction will highly likely be “NOT *X*”. (This hypothesis will be confirmed in chapter 4.3.) This problem becomes more serious with small training dataset [9]. The original HEX model does not cope with such situation, as the potential function (3.1) deactivates a bottom-level node as long as its unary confidence is below the decision boundary.

Chapter 4

Reimplementation

4.1 Dataset

The original HEX model is reimplemented as the baseline of this work. While [4] used ILSVRC 2012 dataset [19] for its rich structure in the label space, this work employs PASCAL VOC 2012 [7] for its simplicity. The complete hierarchical subgraph is shown in figure 4.2. As discussed in chapter 1, the two datasets are designed for different tasks: ILSVRC is an exclusive multiclass classification problem, whereas PASCAL is an independent multiclass classification problem. To adapt to ILSVRC task 1, the train+val dataset of PASCAL is filtered through the following criteria:

1. If there is only one annotated object in the image, the image is labelled to that object.

2. If the largest annotated object in the image is more than twice as large as the second largest one, the image is labelled to its dominating object.

Another major difference between ImageNet and PASCAL is that the former one is a balanced dataset, while the latter is highly unbalanced. For example, after filtering, there are 5,324 images labelled as "person", whereas the second most frequent label "dog" has only 817 instances. To rebalance the dataset, 950 images of "person" are subsampled from the filtered dataset. After preprocessing, the dataset contains 8,473 images in total. These images are then split 3:1:1 into train/validation/test set. The distribution of images across labels is illustrated in figure 4.1.

4.2 Algorithms

For consistency with [4], the underlying unary classifier is based on [11] instead of the current state-of-the-art [20, 21]. The CNN, originally trained on ImageNet, is fine-tuned as an independent multiclass classifier on PASCAL, where each image is projected to a 27-dimensional vector. Note that since the original HEX model has no learning part, it was built as a layer into the CNN, achieving end-to-end learning. In this work, CNN and CRF are implemented separately, such that multiple variants of the original HEX model can be tested with the same underlying unary classifier; and

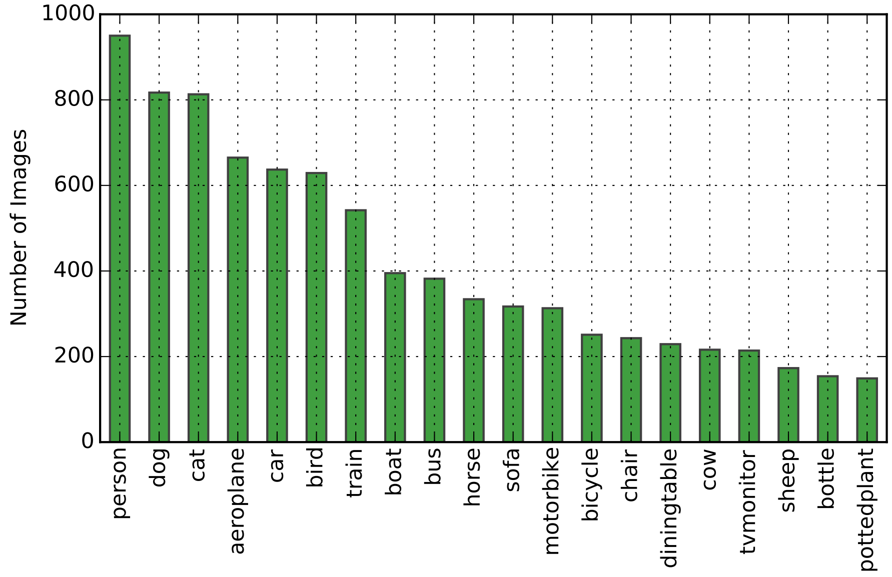

Figure 4.1: Distribution of images according to labels, after preprocessing.

different unary classifiers can be paired with the same variant of HEX model. For details of CNN setup, refer to appendix A.2.

A simpler concept space also lead to the change of inference algorithm. In [4], inference is performed by maximum-a-posteriori (MAP) loopy belief propagation on the HEX graph, where the local state space of each clique is constrained by semantic restrictions (see chapter 5.3 for details). In this work, due to the small concept space of dataset, the inference is performed by calculating the potential function directly for each state in the global state space.

Finally, to guarantee that the benefit of HEX framework is consistent regardless of the underlying unary classifier, different variants of the HEX model are also paired with SVM as the underlying unary classifier. An array of 27 SVMs is trained, each corresponding to a node in the HEX graph. The SVM array accepts the output of the last-but-one layer of [11] as input, assuming that 4,096 neurons provide sufficiently diverse and accurate feature responses *without* fine-tuning on the modified PASCAL dataset. The SVM array predicts distance to decision boundaries, corresponding to raw CNN output without sigmoid transformation. For details of SVM setup, refer to appendix A.2.

4.3 Experiments

In this work, accuracy is tested both in the extended state space and in the original state space, emphasizing on the former one. Testing in the original state space is achieved by limiting the legal state space to assignments with an active bottom-level node in the hierarchical subgraph; or in case of flat classification baseline, to the

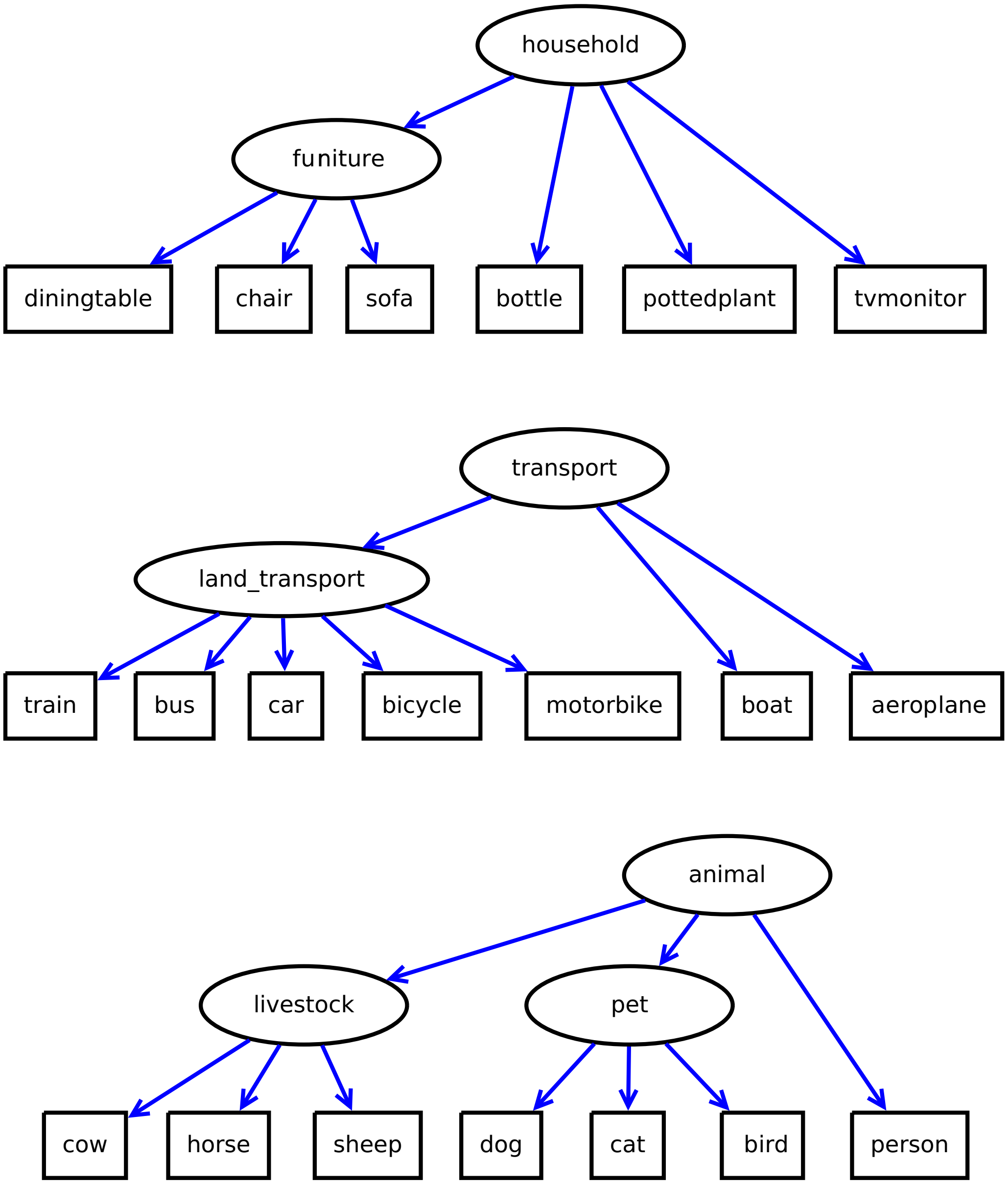

Figure 4.2: The PASCAL dataset has 20 concepts, as shown in rectangles. In this work, these concepts are extended by 7 of their hypernyms, creating a forest of three trees with 27 nodes and 24 hierarchical edges. Exclusive edges are not drawn since they can be implied by the hierarchical subgraph. The state space of this HEX graph has size 28. In comparison, the HEX graph in [4] contains 1000 nodes corresponding to the original concept space, and 820 nodes corresponding to their hypernyms. Note that there is no hierarchical relationship within the original concept space.

original concept space. Note that this is only valid for datasets where all concepts in the original concept space are bottom-level nodes in the hierarchical subgraph.

The empirical test result of [4] and the reimplemented system are compared in table 4.1. Due to the size of PASCAL dataset, relabelling rate of 95% is not attempted in this work. Observations on the experimental results are listed as follows:

1. Regardless of implementation and inference algorithm, the accuracy drops as the relabelling rate grows. This is within expectation.

2. In the extended state space, independent classification baseline by itself tend to have the lowest accuracy. However, in the original state space, the accuracy of independent classification can sometimes beat the softmax baseline. This is a clear sign that nodes corresponding to abstract concepts in the HEX graph receive higher confidence than bottom-level nodes. With the propagation of confidence discussed in chapter 3.2, the HEX framework is able to recover the classification information on bottom-level nodes to a similar level as the softmax baseline.

3. In [4], the original HEX model managed to beat the softmax baseline under all relabelling rates (except for 0%, on which only the accuracy of softmax baseline is provided). However, with the reimplemented system in the extended state space, the accuracy of HEX falls below the softmax baseline. In addition, as the relabelling rate grows, the accuracy of the reimplemented system drops much faster than in [4]. This is a clear sign of insufficient confidence on bottom-level concepts, as discussed in chapter 3.3. This problem is also inspected in table 4.2.

	0%	50%	90%	95%
Softmax	0.626(0.843)	0.564(0.796)	0.529(0.772)	0.508(0.760)
Independent	N/A	0.210(0.452)	0.093(0.272)	0.056(0.172)
HEX	N/A	0.582(0.808)	0.553(0.794)	0.524(0.772)
Softmax	0.669(0.906)	0.332(0.816)	0.007(0.738)	N/A
Independent	0.171(0.863)	0.004(0.811)	0.000(0.407)	N/A
HEX	0.657(0.893)	0.318(0.869)	0.000(0.838)	N/A
Softmax	0.673(0.906)	0.622(0.896)	0.561(0.871)	N/A
Independent	0.726(0.937)	0.673(0.930)	0.323(0.640)	N/A
HEX	0.720(0.894)	0.685(0.888)	0.502(0.872)	N/A

Table 4.1: Comparison of empirical test result in [4] (top) with the reimplemented system (middle: extended concept space; bottom: original concept space). Performance is reported in accuracy, with top-5 accuracy in the bracket. The softmax baseline is obtained by training a CNN that maps an image to exactly one node in the HEX graph, discarding all hierarchical information. The independent classification baseline is obtained by classifying to the concept with the highest unary confidence.

	0%	50%	90%
diningtable	0.4285 ± 0.3011	0.2490 ± 0.2288	0.0108 ± 0.0116
furniture	0.6460 ± 0.3546	0.6762 ± 0.3321	0.6709 ± 0.3380
household	0.7943 ± 0.3041	0.7989 ± 0.2895	0.7975 ± 0.2963
dog	0.7517 ± 0.3424	0.4533 ± 0.2970	0.0222 ± 0.0209
pet	0.8157 ± 0.3015	0.8047 ± 0.3065	0.8056 ± 0.3067
animal	0.8442 ± 0.2787	0.8448 ± 0.2804	0.8467 ± 0.2775

Table 4.2: Under different relabelling rates, with fully trained CNN, the mean and standard deviation of sigmoid-transformed response on validation images labelled “diningtable” (upper) and “dog” (lower). Note that “household” is a hypernym of “furniture”, which is further a hypernym of “diningtable”; and “animal” is a hypernym of “pet”, which is further a hypernym of “dog”. With 490 instances in the training set, label “dog” is a frequent concept, whereas rare concept “diningtable” has 137 instances in the training set. Clearly, the unary response on abstract concepts are not affected by the relabelling rate, yet on bottom-level concepts, the confidence drops to below the decision boundary. This experiment also shows that CNN learns highly abstract concepts accurately, even for concepts without visual clues such as “household”.

Chapter 5

Improvements

5.1 Algorithms

Based on the original HEX model, this work delivers improvements in three stages, where each stage is built on top of the previous one. The high-level goal of improvements is stated as follows: With little confidence on bottom-level nodes, the classifier should attempt to classify to an assignment with an active bottom-level concept. However, in case the classifier is not confident enough to do so, it should also be allowed to predict to a joint assignment in the extended state space.

In the first stage, the confidence on inactive nodes are taken into consideration, in addition to active ones. The resulting unnormalised potential function is shown as follows:

$$\tilde{p}(y|x) = \prod_{i \in V} \exp\{x_i \cdot I[y_i = 1] + (1 - x_i) I[y_i = 0]\} I[y \text{ legal}] \tag{5.1}$$

where x_i denotes the confidence of node i being true, and $1 - x_i$ for being false. This requires the range of the unary prediction to be within $[0, 1]$, and the decision boundary to be 0.5. To achieve this, unary predictions from the underlying classifier are normalised with the sigmoid function.

Theoretically, stage 1 fix should benefit in two aspects: First, the potential function considers the same number of nodes for every legal joint assignment, therefore fixing the unnormalised depth problem discussed in chapter 3.3. Second, the potential function is able to refer to the confidence of inactive nodes, hence benefiting from those that are far from the decision boundary. However, a massage on (5.1) shows that it is not far from the original potential function (3.1):

$$\begin{aligned}
\tilde{p}(y|x) &= \prod_{i \in V} \exp\{x_i \cdot I[y_i = 1] + (1 - x_i)(1 - I[y_i = 1])\} I[y \text{ legal}] \\
&= \prod_{i \in V} \exp\{(2x_i - 1) I[y_i = 1] + (1 - x_i)\} I[y \text{ legal}] \\
&= \tilde{p}(\varnothing|x) \prod_{i \in V} \exp\{(2x_i - 1) I[y_i = 1]\} I[y \text{ legal}] \qquad (5.2)
\end{aligned}$$

where $\tilde{p}(\varnothing|x)$ denotes the unnormalised potential of assignment $\varnothing$. Since $\tilde{p}(\varnothing|x)$

only depends on x, it can be absorbed into the partition function $\frac{1}{Z(x)}$. Also, $\exp\{(2x_i - 1)I[y_i = 1]\}$ does not affect the rank of $\tilde{p}(y|x)$ from $\exp\{(x_i \cdot I[y_i = 1]\}$ for all y. Therefore, the difference between (3.1) and (5.1) is no more than the sigmoid transformation.

In the second stage, pairwise terms are added to the potential function. This fix is a remedy to the low confidence problem on bottom-level nodes when the relabelling rate is high, by exploiting the fact that the confidence on the immediate parents of bottom-level nodes are not affected by the relabelling rate. In addition, two regularisation constants are added to the potential function, such that the contributions from unary and pairwise terms are balanced. The resulting normalised potential function is shown as follows:

$$\begin{aligned} p(y|x) = \frac{1}{Z(x)} \exp\left\{ \frac{1}{|V|} \sum_{i \in V} x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0] \right\} \\ \cdot \exp\left\{ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} x_i x_j \cdot I[y_i = y_j = 1] \right\} \cdot I[y \text{ legal}] \end{aligned} \tag{5.3}$$

In the third stage, a weighting factor is added to each unary and pairwise term. This weighting factor is learned form all images labelled to bottom-level concepts in the training dataset. Note that such training scheme cannot cover concepts from the original concept space that are not bottom-level nodes in the HEX graph, as it is not possible to distinguish between images that are correctly labelled to that concept, or images that are relabelled to its immediate parents. For example, in figure 3.1, only images labelled to "cat", "Husky", "Labrador", and "person" are used to train the CRF, while images labelled to "dog" are discarded, even though "dog" is within the original concept space. The resulting normalised potential function is shown as follows:

$$\begin{aligned} p_\theta(y|x) = \frac{1}{Z(x)} \exp\left\{ \frac{1}{|V|} \sum_{i \in V} w_i (x_i \cdot I[y_i = 1] + (1 - x_i) \cdot I[y_i = 0]) \right\} \\ \cdot \exp\left\{ \frac{1}{|E_h|} \sum_{(i,j) \in E_h} t_{ij} \cdot x_i x_j \cdot I[y_i = y_j = 1] \right\} \cdot I[y \text{ legal}] \end{aligned} \tag{5.4}$$

where θ is the concatenation of $\{\forall i \in V : w_i\}$ and $\{\forall (i,j) \in E_h : t_{ij}\}$. θ is selected by optimising for log-likelihood:

$$\theta = \arg\min_\theta \left\{ -\frac{C}{N} \log \prod_{(x,y) \in D} p_\theta(y|x) + \frac{1}{2} \|\theta\|^2 \right\} \tag{5.5}$$

where the weighting of regularisation term $\frac{1}{2}\|\theta\|^2$ is determined by C, a constant chosen by cross-validation. In this work, $C = 1000$. The gradient is calculated piecewisely:

$$\nabla_\theta \log \prod_{(x,y)\in D} p_\theta(y|x) = \begin{bmatrix} \nabla_w \log \prod_{(x,y)\in D} p_\theta(y|x) \\ \nabla_t \log \prod_{(x,y)\in D} p_\theta(y|x) \end{bmatrix} = \begin{bmatrix} \sum_{(x,y)\in D} \nabla_w \log p_\theta(y|x) \\ \sum_{(x,y)\in D} \nabla_t \log p_\theta(y|x) \end{bmatrix} \tag{5.6}$$

The log-likelihood in (5.6) is expanded as follows:

$$\log p_\theta(y|x) = \frac{1}{|V|}\sum_{i\in V} w_i(x_i \cdot I[y_i = 1] + (1-x_i)\cdot I[y_i = 0]) + \frac{1}{|E_h|}\sum_{(i,j)\in E_h} t_{ij}\cdot x_i x_j \cdot I[y_i = y_j = 1] - \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \tag{5.7}$$

Therefore, the gradient in the first line of (5.6) is calculated as:

$$\begin{aligned} \nabla_w \log p_\theta(y|x) &= \frac{1}{|V|}\Big[x_i \cdot I[y_i = 1] + (1-x_i)\cdot I[y_i = 0]\Big]_{i\in V} - \nabla_w \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\ &\text{define } \phi_u(x,y) = \frac{1}{|V|}\Big[x_i \cdot I[y_i = 1] + (1-x_i)\cdot I[y_i = 0]\Big]_{i\in V} \\ &= \phi_u(x,y) - \sum_{\hat{y}} p_\theta(\hat{y}|x)\cdot \phi_u(x,\hat{y}) \end{aligned} \tag{5.8}$$

where the gradient of the partition function in (5.8) is calculated as:

$$\begin{aligned} \nabla_w \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) &= \frac{1}{\sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x)} \nabla_w \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\ &= \frac{1}{Z(x)} \sum_{\hat{y}} \nabla_w \exp\left\{\frac{1}{|V|}\sum_{i\in V} w_i(x_i\cdot I[y_i = 1] + (1-x_i)\cdot I[y_i = 0])\right\} \\ &\quad \cdot \exp\left\{\frac{1}{|E_h|}\sum_{(i,j)\in E_h} t_{ij}\cdot x_i x_j \cdot I[y_i = y_j = 1]\right\} \\ &= \frac{1}{Z(x)} \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \cdot \frac{1}{|V|}\Big[x_i \cdot I[\hat{y}_i = 1] + (1-x_i)\cdot I[\hat{y}_i = 0]\Big]_{i\in V} \\ &= \sum_{\hat{y}} p_\theta(\hat{y}|x)\cdot \phi_u(x,\hat{y}) \end{aligned} \tag{5.9}$$

Similarly, the gradient in the second line of (5.6) is calculated as:

$$\begin{aligned}\nabla_t \log p_\theta(y|x) &= \underbrace{\frac{1}{|E_h|}\Big[t_{ij}\cdot x_i x_j \cdot I[y_i = y_j = 1]\Big]_{(i,j)\in E_h}}_{\phi_t(x,y)} - \nabla_t \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\ &= \phi_t(x,y) - \sum_{\hat{y}} p_\theta(\hat{y}|x)\cdot \phi_t(x,\hat{y}) \end{aligned} \tag{5.10}$$

where the gradient of the partition function in (5.10) is calculated as:

$$\begin{aligned}\nabla_t \log \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) &= \frac{1}{\sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x)} \nabla_t \sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x) \\ &= \frac{1}{Z(x)} \sum_{\hat{y}} \nabla_t \exp\left\{\frac{1}{|V|}\sum_{i\in V} w_i\big(x_i \cdot I[y_i = 1] + (1 - x_i)\cdot I[y_i = 0]\big)\right\} \\ &\quad \cdot \exp\left\{\frac{1}{|E_h|}\sum_{(i,j)\in E_h} t_{ij}\cdot x_i x_j \cdot I[y_i = y_j = 1]\right\} \\ &= \frac{1}{Z(x)}\sum_{\hat{y}} \tilde{p}_\theta(\hat{y}|x)\cdot \frac{1}{|E_h|}\Big[t_{ij}\cdot x_i x_j \cdot I[\hat{y}_i = \hat{y}_j = 1]\Big]_{(i,j)\in E_h} \\ &= \sum_{\hat{y}} p_\theta(\hat{y}|x)\cdot \phi_t(x,\hat{y}) \end{aligned} \tag{5.11}$$

The derivation above satisfies the general property that for a CRF whose potential function in the exponential family, the gradient of the log partition function is the expectation of a feature function. Note that since weights represent the credibility of unary and pairwise confidence, all entries of θ are non-negative.

5.2 Experiments

With CNN as the underlying unary classifier, the overall performance of different variants of the HEX model are compared in table 5.1. These performance are broken down to concepts in table 5.2, 5.3, and 5.4. Table 5.5 contains a summary of the response of CNN on validation images of different concepts. The unary and pairwise weightings learned in stage 3 are shown in table 5.6 and 5.7. For equivalent results with SVM as the underlying unary classifier, refer to appendix A.3. Observations on the experimental results are listed as follows:

1. As a confirmation to (5.2), the difference between the accuracy of stage 1 and the original HEX model is minor.

2. In the extended state space, stage 2 allows a significant accuracy enhancement under all relabelling rates. The largest enhancement is under 50% relabelling rate, where the accuracy of stage 2 leads the original HEX model by 14.49 percentage points. In addition, such improvement is consistent over all concepts,

as shown in the breakdown of accuracy in table 5.2, 5.3, and 5.4. This is a clear sign that the relationships considered in the original HEX model are too weak. Hence, a potential direction for further improvement is to incorporate more relationships.

3. Under 50% relabelling rate, the overall accuracy of stage 2 and 3 are very close. However, as shown in table 5.3, broken down to concepts, their accuracy are very different. Compared to stage 2, stage 3 is slightly inferior on most bottom-level concepts; but has an advantage of over 10 percentage points on concept "bus", "bicycle", "horse", and "sheep". The learned unary weight on these concepts are zero, as shown in table 5.6. Note that in such case, the confidence of a node is still considered in the pairwise term of the potential function. No clear trend has been found regarding why the unary weights of these concepts are set to zero during optimisation.

4. Under 90% relabelling rate, stage 3 is the only system to achieve accuracy higher than 10% in the extended state space. Such improvement of overall accuracy is entirely the result of concept "bus", "bicycle", "horse", "sheep", and "bird", on which the learned unary weights are zero. The accuracy on other concepts are zero.

	0%	50%	90%
Softmax	0.6698(0.8580)	0.3325(0.7161)	0.0112(0.5635)
Independent	0.1716(0.7339)	0.0047(0.6258)	0.0000(0.1728)
Original HEX	0.6573(0.7779)	0.3182(0.7193)	0.0000(0.4774)
Stage 1	0.6579(0.7790)	0.3182(0.7197)	0.0000(0.5154)
Stage 2	0.6923(0.7969)	0.4631(0.7470)	0.0005(0.5908)
Stage 3	0.6882(0.7802)	0.4643(0.6496)	0.1454(0.3646)
Softmax	0.6739(0.8580)	0.6223(0.8301)	0.5617(0.7909)
Independent	0.7268(0.8889)	0.6739(0.8705)	0.3230(0.5385)
Original HEX	0.7209(0.8788)	0.6852(0.8574)	0.5029(0.7838)
Stage 1	0.7214(0.8622)	0.6799(0.8337)	0.5000(0.7850)
Stage 2	0.7238(0.7957)	0.6252(0.7470)	0.3129(0.5908)

Table 5.1: Comparison of empirical test result of different variants of the HEX model (rows) under different relabelling rates (columns), in the extended state space (upper) and the original one (lower). Performance is reported in accuracy, with top-3 accuracy in the bracket (in contrast to top-5 accuracy in table 4.1). Note that stage 3 improvement only applies to the extended state space.

Concept	Softmax	Original HEX	Stage 2	Stage 3	Stage 2− Original	Stage 3− Stage 2
diningtable	0.4821	0.5357	0.5535	0.5535	0.0178	0.0000
chair	0.4666	0.3111	0.3777	0.3555	0.0666	-0.0222
sofa	0.4310	0.3275	0.3793	0.3620	0.0518	-0.0173
bottle	0.6000	0.5666	0.6333	0.5333	0.0667	-0.1000
pottedplant	0.3928	0.3928	0.4642	0.4642	0.0714	0.0000
tvmonitor	0.8500	0.6250	0.6250	0.5500	0.0000	-0.0750
train	0.8166	0.8666	0.8916	0.8750	0.0250	-0.0166
bus	0.6764	0.6764	0.7058	0.7647	0.0294	0.0589
car	0.8281	0.7343	0.7656	0.7500	0.0313	-0.0156
bicycle	0.8148	0.8148	0.8148	0.7962	0.0000	-0.0186
motorbike	0.8035	0.6785	0.7142	0.6964	0.0357	-0.0178
aeroplane	0.8137	0.7724	0.8000	0.7931	0.0276	-0.0069
boat	0.6301	0.6986	0.6986	0.6986	0.0000	0.0000
cow	0.3269	0.3846	0.4615	0.4423	0.0769	-0.0192
horse	0.4057	0.3768	0.4492	0.5072	0.0724	0.0580
sheep	0.3529	0.3529	0.4411	0.5588	0.0882	0.1177
dog	0.7272	0.7954	0.8181	0.8068	0.0227	-0.0113
cat	0.5793	0.5655	0.6000	0.5862	0.0345	-0.0138
bird	0.6886	0.7169	0.8018	0.7924	0.0849	-0.0094
person	0.7313	0.7213	0.7412	0.7363	0.0199	-0.0049
household	N/A	0.7859	0.7859	0.7548	0.0000	-0.0311
furniture	N/A	0.6477	0.6666	0.6603	0.0189	-0.0063
transport	N/A	0.9565	0.9565	0.9658	0.0000	0.0093
landtransport	N/A	0.9413	0.9483	0.9483	0.0070	0.0000
animal	N/A	0.8888	0.8876	0.8863	-0.0012	-0.0013
livestock	N/A	0.5741	0.6000	0.6000	0.0259	0.0000
pet	N/A	0.8594	0.8758	0.8735	0.0164	-0.0023

Table 5.2: With CNN as the underlying unary classifier, comparison of accuracy of variants of the HEX model (columns) broken down to concepts. The relabelling rate is 0%, and the classification is performed in the extended state space. The rightmost two columns contain the difference between the accuracy of stage 2 and original, and the difference between the accuracy of stage 3 and 2, respectively. Note that the accuracy of the softmax baseline is not available for abstract concepts (below the horizontal line), as all images are labelled to bottom-level concepts in PASCAL.

Concept	Softmax	Original	Stage 2	Stage 3	Stage 2— Original	Stage 3— Stage 2
diningtable	0.2321	0.3571	0.4821	0.4642	0.1250	-0.0179
chair	0.0888	0.0666	0.1333	0.1111	0.0667	-0.0222
sofa	0.1206	0.0689	0.2068	0.1379	0.1379	-0.0689
bottle	0.3000	0.1666	0.2666	0.3000	0.1000	0.0334
pottedplant	0.1071	0.0357	0.0357	0.1071	0.0000	0.0714
tvmonitor	0.5250	0.4500	0.4750	0.4750	0.0250	0.0000
train	0.1583	0.5000	0.6583	0.5666	0.1583	-0.0917
bus	0.0294	0.5000	0.6911	0.8235	0.1911	0.1324
car	0.0703	0.3125	0.4921	0.4531	0.1796	-0.039
bicycle	0.1481	0.3333	0.4629	0.8518	0.1296	0.3889
motorbike	0.3214	0.3750	0.5000	0.4821	0.1250	-0.0179
aeroplane	0.5034	0.3586	0.6000	0.4551	0.2414	-0.1449
boat	0.2602	0.1780	0.3150	0.2739	0.1370	-0.0411
cow	0.1153	0.0384	0.1153	0.0576	0.0769	-0.0577
horse	0.1304	0.1304	0.2028	0.6956	0.0724	0.4928
sheep	0.2352	0.1470	0.2352	0.5000	0.0882	0.2648
dog	0.6363	0.3863	0.5738	0.5056	0.1875	-0.0682
cat	0.4137	0.2275	0.3724	0.3241	0.1449	-0.0483
bird	0.5000	0.4056	0.5660	0.5566	0.1604	-0.0094
person	0.5373	0.4328	0.5572	0.5373	0.1244	-0.0199
household	N/A	0.7859	0.7743	0.7626	-0.0116	-0.0117
furniture	N/A	0.6729	0.6981	0.6729	0.0252	-0.0252
transport	N/A	0.9534	0.9565	0.9549	0.0031	-0.0016
landtransport	N/A	0.9201	0.9413	0.9389	0.0212	-0.0024
animal	N/A	0.8825	0.8786	0.8735	-0.0039	-0.0051
livestock	N/A	0.5419	0.6129	0.6967	0.071	0.0838
pet	N/A	0.8337	0.8548	0.8548	0.0211	0.0000

Table 5.3: Comparison of accuracy broken down to concepts, under 50% relabelling rate, in the extended state space, with CNN as the underlying unary classifier. Note that the accuracy on abstract concepts are consistently high for all models, a clear sign of sufficiently high confidence on nodes corresponding to abstract concepts.

Concept	Softmax	Original	Stage 2	Stage 3	Stage 2− Original	Stage 3− Stage 2
diningtable	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
chair	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
sofa	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
bottle	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
pottedplant	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
tvmonitor	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
train	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000
bus	0.0147	0.0000	0.0000	0.8382	0.0000	0.8382
car	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
bicycle	0.0000	0.0000	0.0000	0.8148	0.0000	0.8148
motorbike	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
aeroplane	0.0137	0.0000	0.0000	0.0000	0.0000	0.0000
boat	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
cow	0.0192	0.0000	0.0000	0.0000	0.0000	0.0000
horse	0.0144	0.0000	0.0000	0.6521	0.0000	0.6521
sheep	0.0000	0.0000	0.0000	0.2352	0.0000	0.2352
dog	0.0397	0.0000	0.0000	0.0000	0.0000	0.0000
cat	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
bird	0.0188	0.0000	0.0094	0.8584	0.0094	0.8490
person	0.0199	0.0000	0.0000	0.0000	0.0000	0.0000
household	N/A	0.8093	0.7548	0.7548	-0.0545	0.0000
furniture	N/A	0.6981	0.7044	0.6666	0.0063	-0.0378
transport	N/A	0.9580	0.9534	0.9534	-0.0046	0.0000
landtransport	N/A	0.9413	0.9342	0.9319	-0.0071	-0.0023
animal	N/A	0.8403	0.8799	0.8748	0.0396	-0.0051
livestock	N/A	0.6064	0.6451	0.7096	0.0387	0.0645
pet	N/A	0.8337	0.8665	0.8711	0.0328	0.0046

Table 5.4: Comparison of accuracy broken down to concepts, under 90% relabelling rate, in the extended state space, with CNN as the underlying unary classifier. Note that a concept receives a positive accuracy in stage 3 if and only if its unary weight is set to zero during optimisation.

Concept	0%	50%	90%
diningtable	0.4285 ± 0.3011	0.2490 ± 0.2288	0.0108 ± 0.0116
chair	0.4238 ± 0.3035	0.1375 ± 0.1221	0.0068 ± 0.0052
sofa	0.4466 ± 0.3345	0.2191 ± 0.2027	0.0083 ± 0.0065
bottle	0.6116 ± 0.3785	0.3013 ± 0.3196	0.0097 ± 0.0076
pottedplant	0.4008 ± 0.3749	0.1218 ± 0.1016	0.0061 ± 0.0061
tvmonitor	0.7321 ± 0.3329	0.4790 ± 0.3238	0.0304 ± 0.0246
train	0.8513 ± 0.2687	0.5044 ± 0.2880	0.0046 ± 0.0044
bus	0.8517 ± 0.2642	0.5511 ± 0.2253	0.0094 ± 0.0083
car	0.7582 ± 0.3518	0.3866 ± 0.2717	0.0109 ± 0.0105
bicycle	0.6957 ± 0.3544	0.3214 ± 0.2975	0.0146 ± 0.0130
motorbike	0.7285 ± 0.3231	0.3698 ± 0.2470	0.0075 ± 0.0050
aeroplane	0.7812 ± 0.3220	0.4535 ± 0.2675	0.0062 ± 0.0082
boat	0.7689 ± 0.3253	0.3321 ± 0.2862	0.0072 ± 0.0067
cow	0.4174 ± 0.2945	0.0924 ± 0.0833	0.0047 ± 0.0053
horse	0.4890 ± 0.3544	0.2449 ± 0.2319	0.0055 ± 0.0042
sheep	0.4635 ± 0.3312	0.2753 ± 0.2502	0.0042 ± 0.0035
dog	0.7517 ± 0.3424	0.4533 ± 0.2970	0.0222 ± 0.0209
cat	0.6678 ± 0.3688	0.3223 ± 0.2397	0.0306 ± 0.0416
bird	0.7499 ± 0.3318	0.4043 ± 0.2510	0.0619 ± 0.1114
person	0.7056 ± 0.3563	0.4345 ± 0.2914	0.0312 ± 0.0243
household	0.7943 ± 0.3041	0.7989 ± 0.2895	0.7975 ± 0.2963
furniture	0.6460 ± 0.3546	0.6762 ± 0.3321	0.6709 ± 0.3380
transport	0.9402 ± 0.1713	0.9277 ± 0.1879	0.9324 ± 0.1844
landtransport	0.9274 ± 0.1925	0.9169 ± 0.1946	0.9185 ± 0.2047
animal	0.8442 ± 0.2787	0.8448 ± 0.2804	0.8467 ± 0.2775
livestock	0.6514 ± 0.3555	0.6260 ± 0.3626	0.6530 ± 0.3550
pet	0.8157 ± 0.3015	0.8047 ± 0.3065	0.8056 ± 0.3067

Table 5.5: Under different relabelling rates, with fully trained CNN, the mean and standard deviation of sigmoid-transformed response on validation images of different concepts. Note that under high relabelling rate, the unary confidence on bottom-level concepts are far on the negative side of the decision boundary, in this case 0.5.

Node	0%	50%	90%	90%−0%
diningtable	2.1692	2.1832	2.2378	0.0686
chair	1.4113	1.3210	1.2302	-0.1811
sofa	2.1080	2.1596	2.2351	0.1271
bottle	0.6178	0.6195	0.9062	0.2884
pottedplant	0.8744	0.9273	0.8470	-0.0274
tvmonitor	0.4812	0.7008	0.5482	0.0670
train	2.0026	2.1006	2.2922	0.2896
bus	0.0000	0.0000	0.0000	0.0000
car	1.0358	1.1823	1.2010	0.1652
bicycle	0.0000	0.0000	0.0000	0.0000
motorbike	2.1447	2.1935	2.2931	0.1484
aeroplane	2.0975	2.1355	2.3083	0.2108
boat	1.5493	1.7257	1.7288	0.1795
cow	1.5618	1.5908	1.5191	-0.0427
horse	0.0000	0.0000	0.0000	0.0000
sheep	0.0000	0.0000	0.0000	0.0000
dog	1.8704	2.0003	2.2625	0.3921
cat	1.9724	2.0752	2.2553	0.2829
bird	3.7825	2.6603	0.0000	-3.7825
person	1.3138	1.3928	1.6450	0.3312
household	0.6735	0.7853	0.8827	0.2092
furniture	1.7698	1.6832	1.8698	0.1000
transport	0.1756	0.3454	0.9849	0.8093
landtransport	2.9717	2.9475	3.3086	0.3369
animal	2.6498	2.4981	3.0661	0.4163
livestock	0.0000	0.0000	0.0000	0.0000
pet	4.0644	3.9751	4.0960	0.0316

Table 5.6: Unary weights learned in stage 3, with CNN as the underlying classifier.

Edge	0%	50%	90%	90%−0%
(household, bottle)	0.9686	0.9846	0.9974	0.0288
(household, pottedplant)	1.0023	1.0215	1.0097	0.0074
(household, tvmonitor)	0.9444	0.9575	0.9947	0.0503
(household, furniture)	0.4723	0.5110	0.5250	0.0527
(furniture, diningtable)	0.9642	0.9699	0.9981	0.0339
(furniture, chair)	0.9579	0.9891	1.0003	0.0424
(furniture, sofa)	0.9367	0.9622	0.9981	0.0614
(transport, aeroplane)	0.8654	0.8935	0.9963	0.1309
(transport, boat)	0.9185	0.9540	0.9972	0.0787
(transport, landtransport)	2.2961	2.2119	2.4417	0.1456
(landtransport, train)	0.8297	0.8858	0.9977	0.1680
(landtransport, bus)	1.0487	1.1213	1.0210	-0.0277
(landtransport, car)	0.8752	0.9193	1.0055	0.1303
(landtransport, bicycle)	0.9760	0.9812	1.0009	0.0249
(landtransport, motorbike)	0.9074	0.9335	0.9972	0.0898
(animal, person)	0.8808	0.9133	0.9949	0.1141
(animal, livestock)	0.5204	0.5164	0.5153	-0.0051
(animal, pet)	1.8062	1.7930	1.9839	0.1777
(livestock, cow)	0.9623	0.9867	0.9993	0.0370
(livestock, horse)	0.9450	0.9645	0.9993	0.0543
(livestock, sheep)	0.9741	0.9828	0.9997	0.0256
(pet, dog)	0.7774	0.8477	0.9897	0.2123
(pet, cat)	0.8273	0.8884	0.9816	0.1543
(pet, bird)	3.2935	2.7109	1.3539	-1.9396

Table 5.7: Pairwise weights learned in stage 3, with CNN as the underlying classifier.

5.3 Scalability

Since the improvements of this work are tested on a modified version of the PASCAL dataset instead of ImageNet, a natural question is how scalable this work is. In [4], the HEX graph is processed in two directions before inference: Sparsification, which removes all redundant edges[1] in the HEX graph; and densification, which adds all possible redundant edges to the HEX graph. The state space of the HEX graph is calculated with a recursive algorithm on the maximally densified graph. With the global state space calculated, the inference is performed by MAP loopy belief propagation on the maximally sparsified HEX graph, where the local state space of each clique is constrained by the global state space. The junction tree algorithm hence exploits both the small tree width of the maximally sparsified graph, and the small state space of the maximally densified graph. This inference algorithm is still applicable to the improved potential functions in this work. On the other hand, the CRF learning in stage 3 is not scalable due to to the high-order terms in (3.2).

5.4 Relationship to Probabilistic HEX

One year after [4], the authors of the original work extended the original HEX model to probabilistic HEX [5] by relaxing the hard semantic constraints to probabilistic relations. Such design is theoretically beneficial when uncertainty exists in the relationship between concepts. The pHEX model also unified hierarchical and exclusive edges, hence converting the HEX graph into an Ising Model. This allows existing off-the-shelf inference algorithms to be used, in contrast to the custom inference algorithm discussed in chapter 5.3.

This work and the pHEX model are extensions to the original HEX model in different directions. The major contribution of this work is in coping with dataset imbalance at high relabelling rate. The hard semantic relationship between concepts remains unaffected. Note that stage 3 improvement of this work also relaxed the weights on every nodes and edges in the HEX graph to a learnable value. However, this is different to [5], where the Ising coefficient, chosen by cross-validation, applies uniformly to all hierarchical and semantic edges.

[1] A directed (hierarchical) edge $(a \to b)$ is defined as redundant if there exists a different path from a to b; an undirected (exclusive) edge (a, b) is defined as redundant if there exists an undirected edge (a', b'), such that a' is an ancestor of a (including a itself), and b' is an ancestor of b.

Chapter 6

Conclusion and Future Work

This work is an extension to [4]. While the original work laid down the HEX framework based on solid theoretical foundations, this work presented its rich details, and managed to identify its weakness under the realistic labelling assumption. The major contribution of this work is to allow the HEX model to accurately classify to bottom-level concepts in the HEX graph, when the underlying unary classifier can only provide very low confidence on bottom-level nodes. The cause of such low confidence is either small dataset (absolute lack of training data on bottom-level nodes) or high relabelling rate (relative lack of training data). The improvement over the original HEX model is achieved by incorporating pairwise terms to the potential function of CRF, and allowing learning on the credibility of unary and pairwise terms. The advantage of the improved system over the baseline and the original HEX model has been confirmed empirically on a modified version of the PASCAL dataset.

While is not the major concern, this work also explored effective underlying unary classifiers to support the HEX framework. Based on a CNN [11] fully trained on ImageNet, this work has confirmed that CNN is able to learn very abstract concepts accurately by fine-tuning on a hierarchically-labelled dataset, even for concepts without visual clues. In addition, a reliable unary classifier is also available by combining the feature response extracted from an existing CNN with an SVM array. Compared to using a fine-tuned CNN as the underlying unary classifier, the SVM array can be trained in parallel, hence significantly reducing the time to adapt the HEX framework to a new dataset.

Due to time constraints, the improvements in this work has not been empirically tested on larger datasets such as ImageNet. This is left as future work. In addition, as it has been confirmed, the relationships considered in the original HEX model are too weak. Hence, a potential direction for further improvement is to incorporate more relationships. For example, to add a pairwise term on hierarchical edge $(i \to j)$ where $y_i = 1, y_j = 0$, corresponding to deactivating a more concrete concept.

Appendix A

Supplementary Materials

A.1 HEX Fails with Positive Unary Inputs

Denote the set of active nodes in a state by $\mathcal{S}$. Then there are three cases:

1. Assume there exists exactly one node t, such that $\forall s \in \mathcal{S} : s \to t$. In other words, t is the only node among all active ones with no out-edges. If t has children, then activating any of them improves (3.1).
2. Assume there exists two active nodes with no out-edges. Denote them by t_1 and t_2. Then they must share a common descendant, as otherwise they would be connected by an exclusive edge. Then there are two further cases:
   (a) If t_1 and t_2 share a common immediate child c, then activating c improves (3.1).
   (b) If they only share a distant descendant, then the state of t_1 and t_2 are independent. The problem is thus equivalent to two case 1 in parallel, until case 2.1 is hit.
3. There exists more than two active nodes with no out-edges. then each pair of them must have a common descendant, and the case is equivalent to pairwise case 2.

Hence, it is guaranteed that for positive unary inputs, the original HEX model activates a bottom-level node in the hierarchical subgraph.

A.2 CNN and SVM Setup

The fine-tuning of the CNN from [11] is implemented using Caffe [10]. The final layer of the network is downsized from 1000 neurons to 27, each corresponding to a node in figure 4.2. Since the CNN is fine-tuned as an independent multiclass classifier, the output from the last layer is transformed with the sigmoid function; and the loss is calculated as the Euclidean distance between prediction and the ground truth vector.

The global learning rate is set to 5×10^{-5}. This is significantly lower than in [11], as all layers except the last are assumed well-trained. The global learning rate decreases linearly by 10^{-5} every 5,000 iterations; and the training process terminates after 50,000 iterations. The learning rate of the last layer is set to 5 times the global learning rate. A snapshot is taken every 5,000 iterations, allowing 10 models for selection during cross-validation stage. With a GeForce GTX 470 graphical card, the fine-tuning process finishes within 5 hours on each variant of dataset.

The SVM unary classifier is implemented in Python using `sklearn.svm.SVC` [16], which is a wrapper of `LibSVM` [1]. Attempted kernels are linear, polynomial, and RBF, with hyperparameters left as default. The training process for the SVM array finishes within 6 hours on each variant of dataset *without* parallel training.

A.3 Experiments with SVM as Underlying Unary Classifier

Compared to CNN as the underlying unary classifier (refer to table 5.5), the response of SVM is different in that as the relabelling rate grows, the confidence on bottom-level concepts tend to be higher, while the confidence on abstract concepts tend to be lower. Despite the difference, the experimental results demonstrated similar trends as in chapter 5.2.

The overall performance of different variants of the HEX model are compared in table A.1. These performance are broken down to concepts in table A.2, A.3, and A.4. Table A.5 contains a summary of the response of SVM on validation images of different concepts. The unary and pairwise weightings learned in stage 3 are shown in table A.6 and A.7.

	0%	50%	90%
CNN Softmax	0.6698(0.8580)	0.3325(0.7161)	0.0112(0.5635)
Independent	0.1514(0.7226)	0.0142(0.5540)	0.0000(0.3147)
Original HEX	0.6300(0.7612)	0.2761(0.6846)	0.0083(0.5754)
Stage 1	0.6300(0.7612)	0.2761(0.6846)	0.0083(0.5813)
Stage 2	0.6947(0.8010)	0.4982(0.7523)	0.1235(0.6781)
Stage 3	0.6692(0.7678)	0.4530(0.6371)	0.1597(0.4691)
Independent	0.7523(0.8990)	0.6989(0.8711)	0.4946(0.6882)
Original HEX	0.7416(0.8812)	0.7096(0.8515)	0.5861(0.8117)
Stage 1	0.7399(0.8770)	0.7090(0.8456)	0.5855(0.8117)
Stage 2	0.6840(0.8491)	0.5671(0.8176)	0.4168(0.7553)

Table A.1: Comparison of empirical test result of different variants of the HEX model (rows) under different relabelling rates (columns), in the extended state space (upper) and the original one (lower). Performance is reported in accuracy, with top-3 accuracy in the bracket. Note that since the SVM array is naturally an independent multiclass classifier, CNN softmax is kept the a baseline.

Concept	CNN	Original	Stage 2	Stage 3	Stage 2−	Stage 3−
	Softmax	HEX			Original	Stage 2
diningtable	0.4821	0.6250	0.7142	0.6964	0.0892	-0.0178
chair	0.4666	0.2222	0.2666	0.2666	0.0444	0.0000
sofa	0.4310	0.3275	0.4137	0.3793	0.0862	-0.0344
bottle	0.6000	0.4333	0.4666	0.4666	0.0333	0.0000
pottedplant	0.3928	0.4285	0.4642	0.4285	0.0357	-0.0357
tvmonitor	0.8500	0.5500	0.5750	0.6000	0.0250	0.0250
train	0.8166	0.7333	0.8000	0.7583	0.0667	-0.0417
bus	0.6764	0.6470	0.6911	0.7794	0.0441	0.0883
car	0.8281	0.6640	0.7343	0.6953	0.0703	-0.0390
bicycle	0.8148	0.7037	0.7407	0.7962	0.0370	0.0555
motorbike	0.8035	0.6607	0.7500	0.6964	0.0893	-0.0536
aeroplane	0.8137	0.7862	0.8413	0.8000	0.0551	-0.0413
boat	0.6301	0.6712	0.6986	0.6575	0.0274	-0.0411
cow	0.3269	0.4230	0.5576	0.4230	0.1346	-0.1346
horse	0.4057	0.3188	0.4782	0.5072	0.1594	0.0290
sheep	0.3529	0.5588	0.5882	0.6176	0.0294	0.0294
dog	0.7272	0.6988	0.7272	0.7329	0.0284	0.0057
cat	0.5793	0.6896	0.7448	0.6896	0.0552	-0.0552
bird	0.6886	0.8113	0.8679	0.8679	0.0566	0.0000
person	0.7313	0.6119	0.7064	0.6268	0.0945	-0.0796
household	N/A	0.7431	0.7431	0.7237	0.0000	-0.0194
furniture	N/A	0.6226	0.6729	0.6415	0.0503	-0.0314
transport	N/A	0.9208	0.9223	0.9192	0.0015	-0.0031
landtransport	N/A	0.8591	0.8896	0.8920	0.0305	0.0024
animal	N/A	0.9195	0.9284	0.9208	0.0089	-0.0076
livestock	N/A	0.5677	0.6516	0.6451	0.0839	-0.0065
pet	N/A	0.8641	0.9039	0.9016	0.0398	-0.0023

Table A.2: Comparison of accuracy broken down to concepts, under 0% relabelling rate, in the extended state space, with SVM as the underlying unary classifier.

Concept	CNN Softmax	Original HEX	Stage 2	Stage 3	Stage 2− Original	Stage 3− Stage 2
diningtable	0.2321	0.3214	0.6428	0.5357	0.3214	-0.1071
chair	0.0888	0.0222	0.1333	0.1333	0.1111	0.0000
sofa	0.1206	0.0689	0.1896	0.1379	0.1207	-0.0517
bottle	0.3000	0.1333	0.3333	0.3666	0.2000	0.0333
pottedplant	0.1071	0.1785	0.4285	0.3928	0.2500	-0.0357
tvmonitor	0.5250	0.3000	0.4750	0.5250	0.1750	0.0500
train	0.1583	0.3416	0.5583	0.4333	0.2167	-0.1250
bus	0.0294	0.3088	0.5441	0.7647	0.2353	0.2206
car	0.0703	0.2500	0.4687	0.3359	0.2187	-0.1328
bicycle	0.1481	0.2407	0.5370	0.8148	0.2963	0.2778
motorbike	0.3214	0.2857	0.4821	0.3571	0.1964	-0.1250
aeroplane	0.5034	0.3517	0.6137	0.5034	0.2620	-0.1103
boat	0.2602	0.2054	0.5068	0.4109	0.3014	-0.0959
cow	0.1153	0.1153	0.4038	0.2115	0.2885	-0.1923
horse	0.1304	0.1594	0.3333	0.5507	0.1739	0.2174
sheep	0.2352	0.2352	0.4705	0.5882	0.2353	0.1177
dog	0.6363	0.3181	0.5170	0.4147	0.1989	-0.1023
cat	0.4137	0.3241	0.5448	0.4620	0.2207	-0.0828
bird	0.5000	0.3584	0.6603	0.6603	0.3019	0.0000
person	0.5373	0.3283	0.4925	0.4129	0.1642	-0.0796
household	N/A	0.7548	0.7509	0.7237	-0.0039	-0.0272
furniture	N/A	0.6289	0.7106	0.6477	0.0817	-0.0629
transport	N/A	0.9177	0.9208	0.9192	0.0031	-0.0016
landtransport	N/A	0.8521	0.8943	0.8967	0.0422	0.0024
animal	N/A	0.9118	0.9195	0.9144	0.0077	-0.0051
livestock	N/A	0.5935	0.6838	0.6451	0.0903	-0.0387
pet	N/A	0.8477	0.8922	0.8969	0.0445	0.0047

Table A.3: Comparison of accuracy broken down to concepts, under 50% relabelling rate, in the extended state space, with SVM as the underlying unary classifier.

Concept	CNN Softmax	Original HEX	Stage 2	Stage 3	Stage 2− Original	Stage 3− Stage 2
diningtable	0.0000	0.0000	0.2678	0.0892	0.2678	-0.1786
chair	0.0000	0.0000	0.0222	0.0222	0.0222	0.0000
sofa	0.0000	0.0000	0.0172	0.0000	0.0172	-0.0172
bottle	0.0000	0.0000	0.0333	0.0333	0.0333	0.0000
pottedplant	0.0000	0.0357	0.1428	0.2142	0.1071	0.0714
tvmonitor	0.0000	0.0250	0.0750	0.2000	0.0500	0.1250
train	0.0083	0.0083	0.1916	0.0250	0.1833	-0.1666
bus	0.0147	0.0147	0.1911	0.7647	0.1764	0.5736
car	0.0000	0.0000	0.1250	0.0312	0.1250	-0.0938
bicycle	0.0000	0.0000	0.1851	0.8518	0.1851	0.6667
motorbike	0.0000	0.0000	0.0714	0.0178	0.0714	-0.0536
aeroplane	0.0137	0.0068	0.1172	0.0344	0.1104	-0.0828
boat	0.0000	0.0000	0.1232	0.0684	0.1232	-0.0548
cow	0.0192	0.0000	0.0384	0.0000	0.0384	-0.0384
horse	0.0144	0.0000	0.0289	0.4637	0.0289	0.4348
sheep	0.0000	0.0000	0.0588	0.5294	0.0588	0.4706
dog	0.0397	0.0056	0.1477	0.0625	0.1421	-0.0852
cat	0.0000	0.0344	0.1517	0.0758	0.1173	-0.0759
bird	0.0188	0.0094	0.1792	0.1698	0.1698	-0.0094
person	0.0199	0.0099	0.0895	0.0447	0.0796	-0.0448
household	N/A	0.7509	0.7431	0.7081	-0.0078	-0.0350
furniture	N/A	0.6226	0.7044	0.6352	0.0818	-0.0692
transport	N/A	0.9177	0.9192	0.9192	0.0015	0.0000
landtransport	N/A	0.8521	0.8967	0.8896	0.0446	-0.0071
animal	N/A	0.9093	0.9080	0.9118	-0.0013	0.0038
livestock	N/A	0.5870	0.6838	0.6838	0.0968	0.0000
pet	N/A	0.8384	0.8922	0.8899	0.0538	-0.0023

Table A.4: Comparison of accuracy broken down to concepts, under 90% relabelling rate, in the extended state space, with SVM as the underlying unary classifier.

Concept	0%	50%	90%
diningtable	0.4804 ± 0.1966	0.3840 ± 0.1581	0.2409 ± 0.0641
chair	0.3722 ± 0.1553	0.3001 ± 0.1167	0.2227 ± 0.0789
sofa	0.4534 ± 0.2143	0.3379 ± 0.1419	0.2337 ± 0.0793
bottle	0.5479 ± 0.2583	0.3764 ± 0.1434	0.2797 ± 0.1136
pottedplant	0.4661 ± 0.2345	0.3604 ± 0.1689	0.2582 ± 0.0950
tvmonitor	0.5667 ± 0.2350	0.4103 ± 0.1913	0.2304 ± 0.0936
train	0.7275 ± 0.2600	0.4189 ± 0.1779	0.2538 ± 0.0853
bus	0.7250 ± 0.1979	0.4219 ± 0.1837	0.2560 ± 0.0800
car	0.6404 ± 0.2816	0.3763 ± 0.1844	0.2469 ± 0.0846
bicycle	0.6018 ± 0.2739	0.4036 ± 0.1604	0.2349 ± 0.0751
motorbike	0.5953 ± 0.2172	0.4035 ± 0.1642	0.2171 ± 0.1010
aeroplane	0.7229 ± 0.2358	0.4251 ± 0.1835	0.2362 ± 0.1036
boat	0.6580 ± 0.2325	0.4247 ± 0.2081	0.2646 ± 0.0903
cow	0.4227 ± 0.1850	0.3146 ± 0.1479	0.2227 ± 0.0634
horse	0.5085 ± 0.2165	0.3735 ± 0.1467	0.2220 ± 0.0736
sheep	0.4951 ± 0.1858	0.3590 ± 0.1482	0.2332 ± 0.0900
dog	0.6190 ± 0.2490	0.4123 ± 0.1935	0.2568 ± 0.1072
cat	0.6688 ± 0.2459	0.4223 ± 0.1796	0.2707 ± 0.1041
bird	0.7018 ± 0.2342	0.4219 ± 0.1795	0.2703 ± 0.1179
person	0.5841 ± 0.2590	0.3971 ± 0.2080	0.2465 ± 0.1077
household	0.6598 ± 0.2280	0.6598 ± 0.2280	0.6598 ± 0.2280
furniture	0.5707 ± 0.2492	0.5707 ± 0.2492	0.5707 ± 0.2492
transport	0.8052 ± 0.1949	0.8052 ± 0.1949	0.8052 ± 0.1949
landtransport	0.7691 ± 0.2267	0.7691 ± 0.2267	0.7691 ± 0.2267
animal	0.7816 ± 0.2167	0.7816 ± 0.2167	0.7816 ± 0.2167
livestock	0.6098 ± 0.2389	0.6098 ± 0.2389	0.6098 ± 0.2389
pet	0.7394 ± 0.2151	0.7394 ± 0.2151	0.7394 ± 0.2151

Table A.5: Under different relabelling rates, with SVM as the underlying unary classifier, the mean and standard deviation of sigmoid-transformed response (distance to decision boundary) on validation images of different concepts.

Node	0%	50%	90%	90%−0%
diningtable	1.9265	1.9236	1.7988	-0.1277
chair	1.3386	1.2640	1.2966	-0.0420
sofa	1.8851	1.9036	1.8286	-0.0565
bottle	0.6659	0.6271	0.8228	0.1569
pottedplant	0.9247	0.8957	0.9201	-0.0046
tvmonitor	0.4994	0.7050	0.5832	0.0838
train	1.8813	1.9102	1.8070	-0.0743
bus	0.0450	0.0297	0.0224	-0.0226
car	0.9947	1.1058	0.9904	-0.0043
bicycle	0.0000	0.0000	0.0000	0.0000
motorbike	1.9338	1.9510	1.9340	0.0002
aeroplane	1.8529	1.9190	1.8513	-0.0016
boat	1.4451	1.5748	1.4387	-0.0064
cow	1.4513	1.4529	1.3184	-0.1329
horse	0.0000	0.0000	0.0220	0.0220
sheep	0.0000	0.0000	0.0000	0.0000
dog	1.7648	1.8041	1.7249	-0.0399
cat	1.7568	1.8101	1.7346	-0.0222
bird	3.5105	3.1755	3.2677	-0.2428
person	1.2342	1.2741	1.2668	0.0326
household	0.8339	0.9198	0.9864	0.1525
furniture	1.6695	1.6110	1.7586	0.0891
transport	0.4305	0.4510	1.0331	0.6026
landtransport	2.6186	2.5675	2.9159	0.2973
animal	2.5923	2.4176	2.6773	0.0850
livestock	0.0000	0.0000	0.0000	0.0000
pet	3.8558	3.8488	3.9997	0.1439

Table A.6: Unary weights learned in stage 3, with SVM as the underlying classifier.

Edge	0%	50%	90%	90%−0%
(household, bottle)	0.9506	0.9503	0.9371	-0.0135
(household, pottedplant)	1.1450	1.1220	1.1862	0.0412
(household, tvmonitor)	0.9465	0.9477	0.9507	0.0042
(household, furniture)	0.6116	0.6318	0.6151	0.0035
(furniture, diningtable)	0.9365	0.9363	0.9276	-0.0089
(furniture, chair)	0.9657	0.9668	0.9777	0.0120
(furniture, sofa)	0.9170	0.9223	0.9177	0.0007
(transport, aeroplane)	0.8285	0.8477	0.8472	0.0187
(transport, boat)	0.8906	0.8989	0.8841	-0.0065
(transport, landtransport)	2.0117	1.9375	2.0144	0.0027
(landtransport, train)	0.8558	0.8691	0.8663	0.0105
(landtransport, bus)	1.2733	1.2454	1.2599	-0.0134
(landtransport, car)	0.9485	0.9586	1.0058	0.0573
(landtransport, bicycle)	1.0078	1.0127	1.0162	0.0084
(landtransport, motorbike)	0.8892	0.8977	0.9047	0.0155
(animal, person)	0.9056	0.9143	0.8878	-0.0178
(animal, livestock)	0.7380	0.7416	0.8540	0.1160
(animal, pet)	1.8121	1.8469	2.0426	0.2305
(livestock, cow)	0.9483	0.9517	0.9559	0.0076
(livestock, horse)	0.9648	0.9715	0.9677	0.0029
(livestock, sheep)	1.0186	1.0184	1.0406	0.0220
(pet, dog)	0.8137	0.8258	0.8076	-0.0061
(pet, cat)	0.8031	0.8261	0.8021	-0.0010
(pet, bird)	3.0280	2.8567	3.1028	0.0748

Table A.7: Pairwise weights learned in stage 3, with SVM as the underlying classifier.

References

[1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[2] Xiangyu Chen, Xiao-Tong Yuan, Qiang Chen, Shuicheng Yan, and Tat-Seng Chua. Multi-label visual classification with label exclusive context. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 834–841. IEEE, 2011.

[3] Myung Jin Choi, Antonio Torralba, and Alan S Willsky. A tree-based context model for object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):240–252, 2012.

[4] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. Large-scale object classification using label relation graphs. In *Computer Vision–ECCV 2014*, pages 48–64. Springer, 2014.

[5] Nan Ding, Jia Deng, Kevin Murphy, and Hartmut Neven. Probabilistic label relation graphs with ising models. *arXiv preprint arXiv:1503.01428*, 2015.

[6] Mohamed Elhoseiny, Burhan Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2584–2591. IEEE, 2013.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[8] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie. Object categorization using co-occurrence, location and appearance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[9] Haibo He, Edwardo Garcia, et al. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[13] Marcin Marszałek and Cordelia Schmid. Semantic hierarchies for visual object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[14] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[15] Vicente Ordonez, Jia Deng, Yejin Choi, Alexander C Berg, and Tamara Berg. From large scale image categorization to entry-level categories. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2768–2775. IEEE, 2013.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[18] Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 910–917. IEEE, 2010.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[22] Anne-Marie Tousch, Stéphane Herbin, and Jean-Yves Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognition*, 45(1):333–345, 2012.

[23] Lexing Xie, Rong Yan, Jelena Tešić, Apostol Natsev, and John R Smith. Probabilistic visual concept trees. In *Proceedings of the international conference on Multimedia*, pages 867–870. ACM, 2010.

[24] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.