IMAGE CLASSIFICATION INFORMED BY ONTOLOGY-BASED BACKGROUND KNOWLEDGE

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

Student id: 10408224

Department of Computer Science

Contents

Word Count: 1626

List of Tables

List of Figures

Abstract

IMAGE CLASSIFICATION INFORMED BY ONTOLOGY-BASED BACKGROUND KNOWLEDGE

Mirantha Rangara Bernadeen Jayathilaka Senarath Mudalige Don

A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy, 2022

Techniques in computer vision have evolved over the years and seen breakthroughs in the recent past with fully data-driven approaches such as deep neural networks. Although these approaches have shown impressive capabilities when detecting objects in images, they suffer from several shortcomings such as high data dependency and lack of transparency. This is a problem when dealing with applications where data is scarce or transparency is critical to build trust in the decision making process. For example, vision applications in healthcare or self-driving technology, where the direct impact on human life is high.

The aim of this thesis is to study the role of background knowledge when overcoming these limitations in neural network-based vision models. I focus on ontologies to be the source of background knowledge in the investigations because of their superior capabilities in ensuring the consistency of information and their ability to infer new information from existing information. The downstream task of choice during the experimentation is few-shot image classification which is used to evaluate the influence of background knowledge when classifying visual objects with a few examples.

I propose a framework that integrates ontology-based background knowledge with a vision model which has two major components: (1) Concept embeddings that are learnt by capturing symbolic knowledge from an ontology in a continuous vector space. This study investigates methods to represent different properties of an ontology with embeddings. It further designs and applies techniques to measure their quality when representing the knowledge. (2) A vision model that is guided by the learnt embeddings during the training and inference stages. Experiments are carried out to evaluate the informed vision models with several few-shot image classification benchmarks, where they achieve superior performance compared to existing approaches.

The improvement on few-shot learning capabilities of the vision models achieved through the integration of background knowledge manifests a way to overcome the challenge of high data dependency. Moreover, I argue that the use of learnt concept embeddings enhances the transparency of the vision model behaviour as the distribution of the extracted image features is decided by the embedding space. To this end, I further introduce a framework to measure the degree of error during predictions based on the background knowledge used.

This study also discusses the design and construction of suitable ontologies based on the image labels of datasets used for the vision tasks.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Dr. Tingting Mu and Prof. Uli Sattler who granted me the wonderful opportunity to pursue this PhD research. Their constant guidance and encouragement helped me immensely to stay focused during this challenging journey over the last few years.

I would like to thank my mother Ranjani and my father Jayantha for always believing in me and being my strength every step of the way.

I was lucky to have my life partner, my wife Tharani, right by my side throughout this journey who always encouraged me to strive for the best. Also, the recent addition to our lives, little baby Keanu, whose arrival was the biggest motivation for me.

Chapter 1

Introduction

Computer vision has evolved over the years from capabilities of recognising edges and shapes [MH80] to identifying objects in real world images [VDDP18]. Major breakthroughs emerged in the recent past from techniques using deep convolutional neural networks (DCNNs) [KSH12a] that accelerated research on deep learning-based methods [HZRS16a, IHM+16]. Deep learning builds computational models that learn low-dimensional representations of data with multiple levels of abstraction [LBH15]. They can discover complex patterns by generating these representations through multiple layers of computation. DCNNs aim to discern the subtleties of visual objects found in images by extracting relevant features using the different layers of a neural network [QYLC18]. The most suitable parameter values for these layers are found by optimising on an objective function [LTHS88] in an end-to-end fashion.

Currently, variations of DCNNs are proven to be the state-of-the-art in understating visual features from real world images using several benchmarks [DDS+09b, VZ11, XXY+15]. Therefore, these techniques are increasing being used in many critical applications such as segmentation and classification of biological images [NDL+05, SOPH16, EHSES+21] in healthcare, traffic sign classification [SMC12] and detection of pedestrians [SKCL13] in self-driving cars and human bodies and pose detection [TGJ+15, TS14] for security applications. Moreover, face recognition has seen major success with DCNNs [TYRW14], facilitating both government and commercial adoption of the technology. In addition to vision, DCNNs have also shown promise in areas such as natural language understating [CWB+11] and speech recognition [SKM+13]. The hardware needed to implement these algorithms is increasingly becoming cheaper and more efficient, enabling real-time computations in smartphones, cameras and robots.

With the heavy adoption, the robustness of the technology is vital, especially in critical decision making applications in the real world such as healthcare and self-driving cars. But DCNNs are found to have various challenges yet to be addressed [Mar18].

1.1 Limitations of Deep Convolutional Neural Networks

Given large amounts of labelled data and computational resources, deep neural networks might be capable of finding a comprehensive mapping between a set of inputs and outputs [Mar18]. But in practice, vision systems are often required to learn from a finite amount of labelled data. Also, the data could contain imbalances where examples of some visual features are more that others. The brute force approach to learn structures within the data becomes less effective in these cases, leading to several limitations in DCNNs.

The current DCNN-based vision systems depend on large amounts of labelled data to perform a simple task of classification. For example, the approach that popularised the use of DCNNs in vision tasks by Krizhevsky et al. [KSH12b] trained on 1.2 million high-resolution images from the ImageNet [DDS$^+$09b] dataset. Current systems surpassing human performance in board games such as Go and chess, learn from billions of augmented examples generated within their technique [SHM$^+$16]. Further studies [SFH17] share concerns over the ability of convolutional networks to generalise without large number of labelled examples.

Moreover, the generation of accurately annotated data is very expensive [Cro12]. Companies spend millions of dollars when hiring humans to annotate large amounts of data to be used in training vision systems for applications such as self-driving cars [BGC$^+$21]. This high dependency on data has also given rise to unexpected outcomes such as biases in the predictions of vision systems [SC18]. A recent study [BG18] showed how a publicly used face recognition approach was specifically incorrect when it comes to faces of black people. This behaviour was found to be caused by the hidden biases in the data used to train the neural networks.

Humans demonstrate the ability to learn and identify a new concept with a few examples [LUTG17]. If I tell you that a Dalmatian is a white dog with black spots on the body, you would not need millions of pictures of a Dalmatian to identify it the next time you see one. Lake et al. [LUTG17] further states that the ability to learn through abstract relationships rather than explicit instances is even found in newborns.

Thus, the high data dependency of DCNNs is a significant challenge that is yet to be fully overcome.

Another area of concern is the lack of transparency in deep learning-based approaches. Neural networks are often called as 'black boxes' due to this factor [BCR97]. With millions and sometimes even billions of parameters, it becomes hard to interpret the mappings inside DCNNs [Mar18]. Although progress has been made in visualising intermediate representations in nodes and layers of a neural network [QYLC18, SWM17], the system's complete decision-making process still remains opaque. This offers challenges when adopting DCNNs in critical decision making applications such as in healthcare [ABV+20], where the human users demand comprehensive explanations to decisions.

When correlations are learnt by deep learning approaches between sets of inputs and outputs, the features of these sets are seen as 'flat' or non-hierarchical [Mar18]. For example, it is not possible to inform the system that a 'Poodle' is also a 'Dog' and since 'dogs have heads', 'Poodle' should also have a head. This makes it challenging for DCNNs to represent hierarchical relationships between visual objects. Explicit guidance with labelled data is required if abstract relationships between concepts should be represented [LBL18]. On the other hand, the natural language domain addresses the task of capturing concepts from raw text using vector representations learnt for words [MCCD13b] via co-occurrence information [LG14, KSKW15]. I find inspiration from these findings in the approaches proposed in this thesis when improving DCNNs using language-based knowledge.

Recent findings on the effects of adversarial examples on DCNNs [YHZL19, SHS+18] further emphasises the limitations of the learnt representations via deep learning approaches. A study by Goodfellow et al. [GSS14] showed how small perturbations of the inputs could manipulate a trained DCNN to predict an incorrect class label with high confidence. This behaviour reveals that the mappings learnt by the networks are rather superficial and that they lack the generalised understanding of the concept of a given visual object [Mar18].

It can be noted that DCNNs have mainly remained self-contained systems, meaning the representations learnt are only between the provided inputs and the outputs [Mar18]. But in order to tackle challenges of generalisation and abstract hierarchical levels of understanding, the integration of broader knowledge should be investigated

[Ji19a, dSAdOSS18]. Some approaches [LGF16a] argue how DCNNs can seemingly obtain knowledge such as notion of physics by purely dealing with visual features. But what if the known principles of physics could be encoded as background knowledge into the same system? Similarly, can a DCNN learn that both a 'Poodle' and a 'German Shepherd' are also dogs via background knowledge? Can it then learn from just a few visual examples of each class? These questions motivate the goals of this thesis.

1.2 The Role of Background Knowledge

Lake et al. [LUTG17] claim that infants develop an understating of notions such as intuitive physics, intuitive psychology, causation and compositionality from an early age. This helps them to learn about new interactions with their surroundings much faster. Another study [MVRV99] shows how 7-month-old infants can represent, extract and generalise abstract algebraic rules that help them in the task of language acquisition. Intuitively, it is noticeable how we connect and build relationships with the knowledge we already have when understanding new concepts or visuals. During learning, background knowledge plays a significant role in enhancing the learning process.

In addition to the above, recently there has been considerable progress in computer science research around knowledge-based vision systems [Ji19a, dSAdOSS18, PHBB16, WXC04, ACIV17] that further discusses the potential benefit for background knowledge. In [Ji19a], an interesting categorisation of knowledge that can be used as background knowledge is proposed, namely, permanent theoretical knowledge, circumstantial knowledge, subjective experimental knowledge and data knowledge. Although how these categories are formed is debatable, the importance of looking into different forms of knowledge that can be used as background knowledge is identified. The form of knowledge can be based on the considered vision application, as pointed out in [dSAdOSS18], where the authors curate a number of vision tasks along with the forms of knowledge used to inform the learning process. For example, the attempts of using of knowledge in the form of graphs and probabilistic ontologies during an image classification task motivate this study.

Investigations into the use of first-Order Logic (FOL) is prominently seen in several studies [HML$^+$16, RSR15, DRG17] as well. It is presented how logic can facilitate the use of consistent knowledge with the use of reasoning [CFHP17, RGH18]. As shown in [HML$^+$16, DRG17], adaptation of logical knowledge as constraints during the learning process has generated promising results in areas such as sentiment analysis

and named entity recognition. Moreover, there has been a surge in research under the topic of hybrid learning and reasoning systems [vBdBvH+21, VRMG+19] that explore the unification of statistical (data-driven) and symbolic (knowledge-driven) methods, further reinforcing the ideas on integrating external knowledge into neural network-based approaches and its importance. All these findings motivate the goals of this thesis when investigating deeper on the role of background knowledge in vision systems.

1.3 Forms of Background Knowledge

In this study, I choose ontologies to be the source of background knowledge in the investigated approaches. But looking at the existing work, many different forms of knowledge can be identified as potential candidates to use when informing learning-based algorithms. It is important to compare ontologies with the others and distinguish the benefits.

1.3.1 Similar Datasets

I see the approaches of transfer learning [TS10] as instances of using similar data as background knowledge. For example, to improve a DCNN's ability in classifying hand written letters, first the network can be trained to classify a similar dataset of hand-written digits [PY09]. The network parameters learnt during the first task can then be transferred to the second task with a quick fine-tuning of the last layers. Here, the training with the first dataset becomes a form of background knowledge when the same DCNN is then used for a similar task. But the drawback is, none of the limitations discussed in Section 1.1 are addressed in these approaches. The DCNNs still require large amounts of data and any structure present in the input or output spaces is not understood.

1.3.2 Additional Labels

Another set of approaches propose to extend the labels of existing datasets with additional terms and features [Ji19b]. For example, annotate an image of a bird in an image classification task not only with the label 'Bird' but also with others such as 'feathers', 'has beak', 'has wing', 'has leg' etc [Ji19b]. The idea is to let the DCNN figure out an improved mapping to represent the similarities and dissimilarities between classes

using this extra information in the label space. But the limitations around lack of transparency are still seen in these approaches due the disregarding distinction between hierarchical features with others. For example, whole object references such as 'Bird' is treated the same way as part references such as 'has beak' or 'has wing'.

1.3.3 Text

Language modelling has seen great progress recently [SM13], predominantly due to techniques that capture representations of words in high-dimensional vector spaces [MCCD13b], commonly referred to as word embeddings. These capture the co-occurrence information of words in a body of text and seemingly maps words appearing in similar contexts closer together as points in the vector space. These methods have shown the capability of learning from large corpora of text in an unsupervised manner leading to many useful applications [LY18].

The use of word embeddings to inform learning-based approaches has also been investigated in several studies with some promising findings. Frome et al. [FCS$^+$13] obtained word embeddings relevant to the words of image labels and used them as the objectives when mapping the features of respective images extracted via a vision model during training. Then, inference was carried out according to the similarity between the predicted embeddings and the ground truth word embeddings. The approach was shown to produce superior performance in the zero-shot image classification task . It was found to be capable of not only image classification but also in tasks such as zero-shot image retrieval. Norouzi et al. [NMB$^+$13] further simplifies the approach in [FCS$^+$13], showing that knowledge from the word embeddings can be used only in the inference stage to attain the desired image classification performances.

Overall, the impact of background knowledge in the form of word embeddings is clearly positive, but the limitations are twofold. (1) Unclear distinction between 'terms' and 'concepts'. Sometimes the meaning of a word can be ambiguous and used to refer to different concepts in different contexts. For example, take the noun *sorrel*, which stands for a type of plant but can also be used to call a horse with a light reddish-brown coat. Even though the concepts of a plant and a horse are totally different, word embeddings end up representing both of these with one vector. This is an unclear representation of meaning and the challenge comes when unstructured text is a collection of terms rather than a definition of concepts. (2) Word embeddings lack the capability of representing specialisations and partonomies of knowledge in a meaningful way. For example, the word embeddings resulting from information

saying *Poodle is-a Dog* and *Dog hasPart Tail* are both embedded by points representing words such as *Poodle, Dog* and *Tail* lying in close proximity to each other. The close positioning of points does not capture the information that *Poodle* is a subclass of *Dog* or that *Tail* is a part of a *Dog* which can also be part of a *Poodle*. This is not helpful when trying to overcome the challenges discussed in Section 1.1.

1.3.4 Knowledge Graphs

A number of definitions can be found of knowledge graphs depending on their structure, methods of construction and applications [EW16, HBC+21]. The term 'knowledge graph' was mainly popularised by the introduction of Google's Knowledge Graph [Sin12] in 2012, and the idea as been widely adopted both commercially [MCSFL15] and academically [ABK+07]. Any graph-based knowledge representation does not qualify to become a knowledge graph [EW16]. Hogan et al. [HBC+21] define a knowledge graphs as "*a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities*". Sometimes a knowledge graph is even seen as an ontology representing only object-level information [MSG16a].

Several studies investigate the use of knowledge graphs in providing external knowledge to DCNNs [JXLL18a, LHZ+18]. They can be used either when defining the structure of a model and its hyperparameters [MSG16b] or when guiding the loss function during the training stage [VRMG+19]. Knowledge graph embeddings [Ham20], which are often used when informing learning-based models, are again point embeddings which come with limitations similar to word embeddings as discussed above in Section 1.3.3. Furthermore, knowledge graphs are often scaled and curated via automated techniques [KZG+16], that can cause inconsistencies in knowledge such as contradicting information or duplication. But they lack inbuilt mechanisms to check these so that they can be avoided in applications.

1.3.5 Ontologies

Ontologies capture generalised and structured knowledge [MPSP+09a] that can make use of reasoning tools [SMH08] to test knowledge consistency. Reasoning also provides the ability to infer new knowledge from existing knowledge in an ontology [APS14]. Compared to text and knowledge graphs that tend to be loosely defined (Section 1.3.4), ontologies are sets of axioms with well-defined semantics that give

meaning to concepts and the relationships between them. I argue that these factors make an ontology a richer source of background knowledge compared to others.

The existing approaches when using of ontologies to inform DCNN-based vision models are so far scattered. Some studies show the possibility of using logical rules from ontologies as constraints during the learning process [SG16]. Here, the objective of the leaning model is to satisfy a set of axioms. During a semantic image interpretation task [DSG17], the predictions are the objects detected in an input image along with possible relations between them according to the logical constraints. Several others present approaches to represent ontology-based knowledge as embeddings via an intermediate step to generate graphs [CHJR+21] which can then be used in a downstream task. I find that the design space when generating graphs from an ontology is very large, since an ontology can give rise to many types of graphs that form various class hierarchies (e.g., asserted and inferred). Also, some studies combine ontology embeddings with text embedding [GCC+21], proposing further improvements. But I argue that these approaches bring the limitations of word embeddings (Section 1.3.3) back to the learning models.

I identify the superior performance of approaches that integrate background knowledge in the form of embeddings into DCNN-based vision models [FCS+13, NMB+13]. In term of generating ontology-based embeddings, I identify that approaches such as [KLWYH19] that can embed concepts directly using the ontology axioms as input are promising. But there exists a gap in the understating about the combination of these approaches. This realisation motivates this study.

1.4 Few-shot Image Classification

To assess the impact of integrating ontology-based background knowledge to a DCNN-based vision model, I chose few-shot image classification [TWK+20, DCRS19a] as the downstream task in this study.

Few-shot learning in an image classification context focuses on learning the visual features of a class with a very few image examples. The existing approaches in the area can be divided into several categories [CLK+19a]. First is initialisation-based methods, more generally known as meta-learning. Meta-learning aims to find a good initialisation for a vision model, so it can quickly adapt to a new set of image classes

with few examples [FAL17, NAS18, RRS+18a, Jay19]. These approaches are trained on tasks with two sets of image classes, where the support set consists classes with enough examples, followed by a query set with classes with limited examples [FAL17]. Having trained on the support set, the model should find a fitting initialisation so the query set can be learned with the few examples. Similarly, approaches such as [RL16a] try to learn an optimiser that enables few-shot learning. Another study [MY17] brings in external memory into the mechanism when updating the weights of the vision model. Although these approaches achieve considerable performance when learning with few examples, investigations show that they face challenges when there are domain shifts between the background and novel classes [CLK+19a]. But their success in the area and contributed to establishing the benchmarks [VTBE15, WBW+11b] for the task.

Another set of few-shot learning approaches can be named as hallucination-based methods [WGHH18] that involves generative models to learn how to augment the data available. The goal is to learn a generator based on the background classes, so it can hallucinate new examples of the novel classes with data augmentation. One type of these approaches learns to transfer appearance variations of the background classes [HG17] or uses a generative adversarial network (GAN) [CWD+18] model that performs style transfer [ASE17]. Another type directly integrates the generator model to a meta-learning algorithm when performing classification [WGHH18].

The other major category that many few-shot learning approaches fall under is distance metric learning-based methods. These approaches exploit the extracted image feature vector similarities when classifying new input classes [KZS15]. They differ according to the distance metrics and latent image representations used. Approaches such as [VBL+16b] use cosine similarity as the distance metric, while other such as [SSZ17] look at Euclidean distance between classes mean representations. Further studies investigate areas such as relational modules comparing feature maps [SYZ+18a], ridge regression methods [BHTV18a] and the use of graph neural networks [GB17] in distance-based few-shot learning. Some approaches look at predicting the classifier weights for novel classes either using attention-based modules [GK18a, HQDN19] or directly using class features [QBL18]. Also, studies such as [HGP20b] show how preprocessing class features can improve the performance of distance metric learning-based methods. These methods are found to be simpler compared to meta-learning approaches while producing competitive performance [CLK+19a]. I further identify the flexibility these methods offer in extending standard vision architectures to perform few-shot image classification [HZRS16b].

It is important to clarify why the task of few-shot image classification is appropriate for evaluating a vision system informed by ontology-based background knowledge as proposed in this study. As pointed out in Section 1.2, the role of background knowledge is to help overcome the challenges that DCNN-based vision models face such as high data dependency, lack of transparency and generalisation, as discussed in Section 1.1.

Learning with few example images is the primary goal of few-shot image classification and I find that this provides the opportunity to compare the impact of background knowledge integration with other fully data-driven approaches in overcoming the challenge of high data dependency.

Another comparable vision task that could potentially be used for the evaluation of the impact of background knowledge is zero-shot image classification [WYG18a, SGMN13]. Here, the goal is to predict on totally unseen classes where zero examples were provided to train on. I argue that in the case of zero-shot, there can be an element of randomness in the correct predictions made. Hence I do not consider these approaches in this thesis.

In the approaches of few-shot image classification employed in this study, the underlying architecture is based on deep neural networks [HZRS16c]. Deep learning in general, as discussed in Section 1.1, does not perform well with limited learning examples. This raises the question on why it is still considered over other approaches in machine learning to be the base architecture. I argue that the strengths of deep learning techniques are still above other approaches as shown from their successes in tasks with enough learning examples. Hence, if the same strengths were to be utilised effectively in limited data settings, the impact would be higher.

In terms of improving transparency and generalisation, I identify the opportunity to extend the current evaluation frameworks of image classification[1] to further understand the errors and the behaviour of a vision model during the inference stage based on the background knowledge used in the task.

Hence, this study was designed to investigate the impact of ontology-based background knowledge in a vision system performing few-shot image classification.

[1] An extensive list of exiting evaluation settings in few-shot image classification can be found in Appendix A.1

1.5 Contributions of this Thesis

The following are the contributions of this study in the area of informed machine learning with ontology-based background knowledge.

1.5.1 Construction of Ontologies for Image Datasets

One of the challenges I faced during the investigation of ontology-based background knowledge informed image classification is the lack of suitable existing ontologies that can be used with the benchmark image datasets. Hence, this study investigates the construction of four OWL ontologies for the purpose of investigating informed few-shot image classification tasks. Chapter 4 explains the details of this process, where external knowledge resources and dataset annotations were used to obtain information required for the ontology construction.

1.5.2 Learning and Comparing Ontology-based Embeddings

The proposed approaches of this study integrates background knowledge as concept embeddings with a DCNN-based vision model. Here, it is important to investigate how different embedding types can faithfully capture information from ontologies.

In Chapter 4.4, I present three main approaches to learn ontology-based concept embeddings that capture different types of knowledge. Furthermore, a framework for capturing the quality of the learnt concept embeddings is proposed. The constructed ontologies from Chapter 4 are used in the experiments and the resulting embeddings are evaluated on their quality. I further describe the design decisions taken to produce ontology-based concept embeddings that are suitable for the downstream task of image classification.

1.5.3 Improved Error Analysis for Image Classification using Background Knowledge

This study shows how the proposed informed vision models provide the opportunity to analyse the degree of errors during few-shot image classification according to the ontology-based background knowledge used in a given task. In Chapter 5.6, I propose a framework to capture the semantic meaningfulness of errors of a vision model using

ontology-based similarity measures. The chapter further discusses how a suitable similarity measure should be chosen and the insights gained by understating the behaviour of errors in an informed vision model.

1.5.4 Informing Few-shot Image Classification with Embeddings

An overall framework named ViOCE for integrating ontology-based background knowledge to a DCNN-based vision model is presented. Chapter 5 shows how the types of concept embeddings investigated in this study can be used with ViOCE during the training and inference stages of a vision model. The experimental results evaluate the performance of the informed models in the task of few-shot image classification and compares them to other fully learning-based and knowledge informed approaches. Furthermore, the errors of the models during prediction are analysed for their semantic meaningfulness using the method in Chapter 5.6.

1.6 Published work

Below is a list of papers published in peer-reviewed conferences and workshops that contain some of the work presented in this thesis.

- [JMS21b] Jayathilaka, M., Mu, T. and Sattler, U. Towards Knowledge-aware Few-shot Learning with Ontology-based n-ball Concept Embeddings. Accepted at the 20th IEEE International Conference on Machine Learning and Applications (ICMLA 2021). - Related to part of the work presented in Chapters 4.4 and 5.

- [JMS21a] Jayathilaka, M., Mu, T. and Sattler, U. Ontology-based n-ball Concept Embeddings Informing Few-shot Image Classification. In proceedings of the Combination of Symbolic and Sub-symbolic Methods and their Applications workshop (CSSA @ ECML PDKK 2021). - Related to part of the work presented in Chapters 4.4 and 5.

- [JMS20] Jayathilaka, M., Mu, T. and Sattler, U. Visual-semantic embedding model informed by structured knowledge. In proceedings of the 9th European Starting AI Researchers' Symposium 2020 co-located with 24th European Conference on Artificial Intelligence (ECAI 2020). - Related to preliminary work done on Chapter 5.

- [Jay19] Jayathilaka, M., 2019. Enhancing generalization of first-order meta-learning. In proceedings of the 2nd Learning from Limited Labeled Data (LLD) Workshop at the International Conference on Learning Representations (ICLR 2019). - Related to preliminary work done on Chapter 3.

Chapter 2

Preliminaries

In this chapter, the terminology and the existing work used in this study are described.

2.1 Terminology: Ontologies

I summarise the terms regarding ontologies necessary for this work according to [APS16].

An **ontology** is composed of a finite set of axioms that constrain the interpretations of a set of classes. It is a knowledge representation approach that aims to define a set of shared terms of interest in some domain in an expressive manner.

Description Logics (DLs) are a family of formalisms in knowledge representation that has precisely defined semantics. A DL knowledge base can be called an ontology. Many DLs are identified as fragments of first order logic (FOL). A key differenece between DLs and FOL is the variable-free syntax.

The **Web Ontology Language (OWL)** is the World Wide Web Consortium (W3C) standard ontology language for the web that uses Description Logics. In this study, all ontologies are in OWL 2, the latest version of OWL.

A **class** in OWL stands for a set of instances. In DL, a class is called a concept.[1]

The term **atomic classes** denote all the asserted classes in an ontology, e.g., *Person, Dog*. These are also referred to as **named classes** in this study.

A **taxonomy** provides information about generalisation and specialisation relations (sometimes called is-a relations) between classes.

The term **class hierarchy** is used to denote the taxonomy of named classes of an ontology.

[1] I sometimes use the terms class and concept interchangeably in this study.

A **class expression** describes a class using other atomic classes and properties with helper constructors such as conjunctions $\sqcap$, disjunctions $\sqcup$ and existential restrictions $\exists$.

An **individual** can be an instance of a class, e.g., **Bob** which is an instance of *Person*.

A **property** in OWL is a relationship between two individuals, e.g., *hasTail, hasColor*.

Axioms form the basic elements of an ontology that describe the relationships between different classes, individuals and roles.

A set containing all the axioms that describe relationships between classes is called a **TBox**.

Entailment denotes whether an axiom α is entailed by the ontology O, denoted by $O \models \alpha$.

Subsumption a property that states whether a class C is subsumed by a class D according to the ontology, denoted by $C \sqsubseteq D$. Sometimes the terms **subsumer** and **subsumee** are used to denote C and D respectively, along with the term **superclass** for D *subclass* for C.

Disjointness of a class C with a class D is denoted by $C \sqcap D \sqsubseteq \bot$. This means that, if none of the individuals in C overlap with that of D, the $C \sqcap D$ should be nothing.

A **reasoner** is a program which, given a set of asserted axioms as input, can decide entailment queries such as subsumption and disjointness.

A **sibling class** stands for another class E and shares a common subsumer D with C.

A class is **satisfiable** or **consistent** with an ontology if there is a model where it has instances.

2.1.1 Inferred Class Hierarchy (*ICH*)

Assuming all the named classes $\widetilde{O}_C$ are satisfiable with ontology O, *ICH* contains all possible subsumption relations according to the definition of O as shown in Eq. (2.1). The *ICH* will give what is also known as the transitive closure of the class hierarchy of O that infers the shortcut relations, for example $Poodle \sqsubseteq Animal$ if $Poodle \sqsubseteq Dog$ and $Dog \sqsubseteq Animal$. The opposite of transitive closure is transitive reduct, where these shortcuts are not inferred.

$$ICH(O) = \{P \sqsubseteq Q | P \neq Q, P, Q \in \widetilde{O}_C, O \models P \sqsubseteq Q\}. \tag{2.1}$$

2.2 Concept Similarity Measures for Ontologies

I identify two main ontology-based similarity measures in this study that can quantify the similarity between two classes using the class hierarchy and the class expressions of an ontology [APS14].

2.2.1 Path-based Similarity

First is path-based similarity, where the number of steps between two classes is counted along the transitive reduct of the class hierarchy, i.e., without transitive closure shortcuts. This is the most basic method to quantify the closeness among classes in an ontology. Taking the example hierarchy shown in Figure 2.1, it can be seen that the number of steps between the classes *Poodle* and *German_Shepherd* is 2, i.e., *Poodle* to *Dog* and *Dog* to *German_Shepherd*. Hence, path-based similarity determines that the similarity between *Poodle* and *German_Shepherd* is 2.

But in a similar manner, it can be seen that the number of steps between *Canine* and *Fish* is again 2. This implies a drawback of this method since intuitively, *Poodle* and *German_Shepherd* are more similar than *Canine* and *Fish*. It means that classes that appear closely further down the class hierarchy are more similar than those at the top levels. Path-based similarity does not capture this feature.

2.2.2 Feature-based Similarity

Feature-based similarity addresses the drawback mentioned in path-based similarity by considering ontology-based features of a class to compute the similarity between two classes. I make use of two feature-based similarity measures in this study, named atomic similarity and subconcept similarity, that make use of subsumers (more general classes) of a class and its class expression, respectively.

2.2.2.1 Atomic Similarity

Atomic similarity uses the feature of atomic subsumers of a class from the class hierarchy when measuring the similarity between two classes. The set of all subsumers for each class are found.

Let C and D be classes of ontology O. Then the atomic similarity between C and D is computed as follows:

$$Sim(C,D,O) = \frac{|Sub(C,O) \cap Sub(D,O)|}{|Sub(C,O) \cup Sub(D,O)|}, \tag{2.2}$$

where,

$$Sub(C,O) = \{D \in \widetilde{O}_C \,|\, O \models C \sqsubseteq D\}. \tag{2.3}$$

In the above equations, $|M|$ means the cardinality of M. According to Equation 2.2, the atomic similarity between C and D is the ration between the number of common subsumers of C and D and the number of all subsumers of C and D. Here, the subsumers are retrieved from the named classes in the ontology as shown in Equation 2.3, where $\widetilde{O}_C$ represents the named classes in ontology O.

The resulting similarities according to this measure imply that sibling classes become more similar as they go deeper into the class hierarchy. Taking the example shown in Figure 2.1, $Sim(Poodle, German\_Shepherd) = \frac{3}{5}$, and $Sim(Canine, Fish) = \frac{1}{3}$. Now it can be seen that atomic similarity captures the knowledge that $Poodle$ and $German\_Shepherd$ are more similar to each other than $Canine$ and $Fish$.

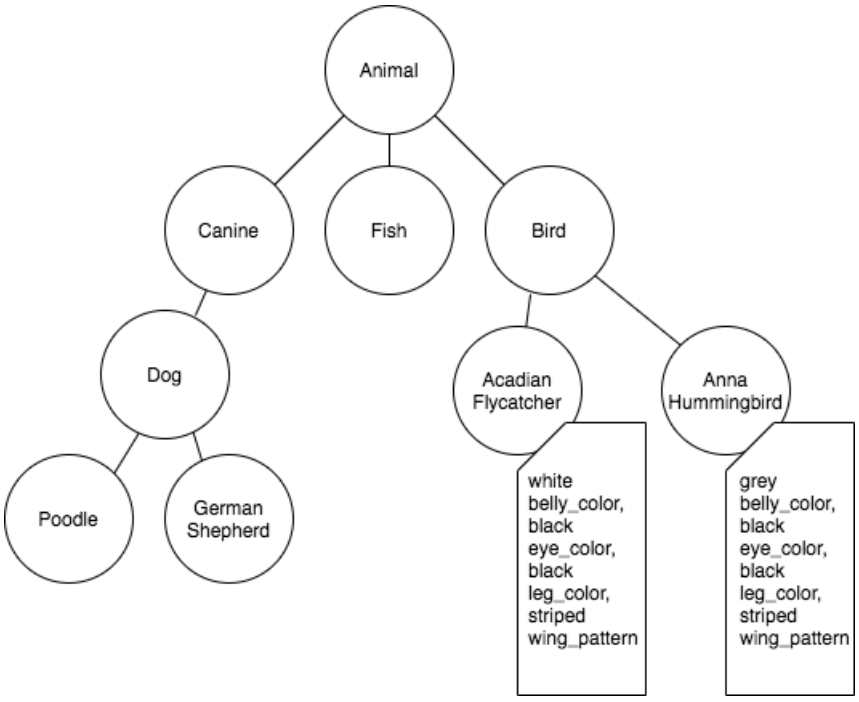


Figure 2.1: An example class hierarchy with named classes and class expressions

2.2.2.2 Subconcept Similarity

Subconcept similarity follows the same technique as atomic similarity, but this time using both atomic subsumers and class expressions as features of the classes. Hence, the definition of $Sub(C, O)$ in Equation 2.2 is modified as follows:

$$Sub(C, O) = \{D \text{ class expression in } O \,|\, O \models C \sqsubseteq D\}. \tag{2.4}$$

Substituting Equation 2.4 in Equation 2.2, the similarity between C and D is computed by the ratio of common subsuming class expression elements of C and D to all class expression elements of C and D.

In Figure 2.1, the example class expressions of $Acadian\_Flycatcher$ and $Anna\_Hummingbird$ define the visual features of each bird species. Considering these, the subconcept similarity between the two classes is $\frac{3}{8}$.

2.3 Embeddings

2.3.1 Word Embeddings

Word embeddings are generated high-dimensional feature vectors that meaningfully characterise words found in a body of text. Most widely used word embeddings capture the distributional semantics of a word using its co-occurrence information with other words, meaning that the embedding generation process is governed by the context that the word appears in.

State-of-the-art word embedding models generally fall under that category of predictive models where they perform predictions given a word and its context [MCCD13a]. Here, two of the popular techniques are namely, Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram models. CBOW model learns by predicting the current word using its context, while Skip-gram model learns by predicting the surrounded words using the current word. Figure 2.2 illustrates these two techniques where $w(t)$ denotes the current word.

In this study, the mentions of word embeddings refer to embeddings that have been generated via a Continuous Skip-Gram model.

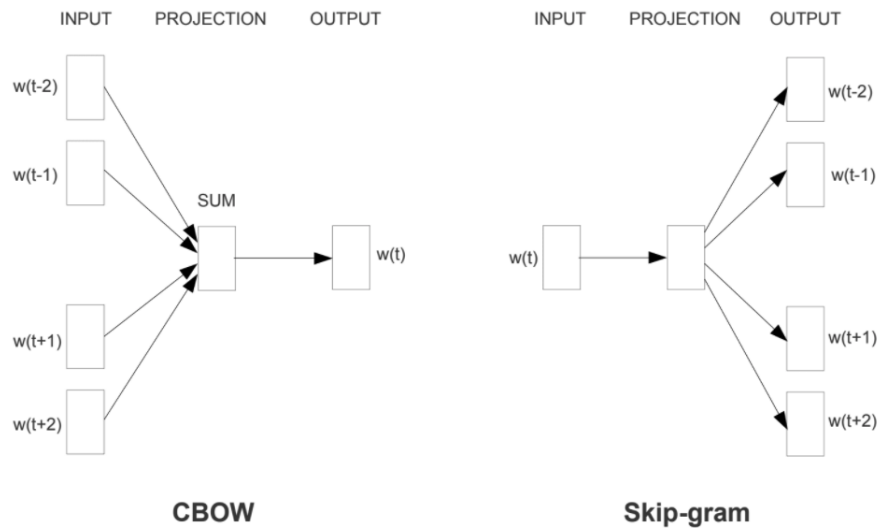


Figure 2.2: Word2Vec learning approaches Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram; taken from [MCCD13a]

2.3.2 Graph-based WordNet Embeddings

Saedi et al. [SBRS18] proposes a technique to compute embeddings for each synset of WordNet [Mil95] using its graph structure. A graph G is converted to an adjacency matrix M such that, if two nodes in G, w_i and w_j (synsets of WordNet in this case) are directly related by an edge, the entry M_{ij} is set to 1 (otherwise 0). Also to account for nodes that are not directly connected to each other, M is further enriched by taking distantly connected nodes and aggregated as shown in Equation 2.5, where n is the length of the path between the two nodes, α (< 1) is a decay factor that determines the effect of path length on M. A longer path between two nodes (larger n) results in a lesser effect on M_G. M_G is normalised using L2-norm and reduced to a set of embeddings with a lower dimensionality using Principal Component Analysis (PCA). Subsequently, each row of M_G corresponding to a synset name of WordNet becomes the embeddings representing that synset.

$$M_G = \sum_{n=0}^{\infty} (\alpha M)^n = (I - \alpha M)^{-1} \tag{2.5}$$

2.3.2.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is used as the dimensionality reduction technique in the above method (Section 2.3.2). The objective of dimensionality reduction

is to generate a set of embeddings $\{\boldsymbol{z}_i\}_{i=1}^n$ of dimension k from a set of samples $\{\boldsymbol{x}_i\}_{i=1}^n$ of dimension d, where $k \ll d$. PCA performs this by mapping feature vectors into smaller number of uncorrelated directions. It extracts a $d \times k$ orthogonal projection matrix P so that the variance of the projected vectors is maximised [MGTA12]:

$$\max_{P \in \mathbb{R}^{d \times k}, P^T P = I_{k \times k}} \frac{1}{n-1} \sum_{i=1}^{n} \left\| P^T \boldsymbol{x}_i - \frac{1}{n} \sum_{j=1}^{n} P^T \boldsymbol{x}_j \right\|_2^2. \tag{2.6}$$

2.3.3 n-balls and EL Embeddings

The notion of ball refers to the volume of space bounded by a sphere and is also called a solid sphere. An n-ball usually refers to a ball in an n-dimensional Euclidean space. The EL embeddings study [KLWYH19] suggests to encode classes as n-balls to ensure that these n-balls respect a set of axioms. Below I explain how it works for encoding subsumption and disjointness as they are the most relevant to our work. Each class P is embedded as an n-ball with its centre denoted by $\boldsymbol{c}_P \in \mathbb{R}^n$ and the radius by $r_P \in \mathbb{R}$. The basic idea is to move one ball inside the other for subsumption and to push two balls to be separated for disjointness. The following loss is minimised to encode an axiom $P \sqsubseteq Q$:

$$\begin{aligned} & l_{P \sqsubseteq Q}(\boldsymbol{c}_P, \boldsymbol{c}_Q, r_P, r_Q) \\ =& \max(0, \|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 + r_P - r_Q - \gamma) \\ &+ \big| \|\boldsymbol{c}_P\|_2 - 1 \big| + \big| \|\boldsymbol{c}_Q\|_2 - 1 \big|, \end{aligned} \tag{2.7}$$

where $\|\cdot\|_2$ denotes the l_2 norm and $\gamma \in \mathbb{R}$ is a user-set hyperparameter. It enforces the inequality $\|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 \leq r_Q - r_P + \gamma$, and regulates the ball centres to be close to a unit sphere. Through controlling the sign of γ, the user can adjust whether to push the P ball completely inside the Q ball. In a similar fashion, the loss for encoding an axiom, $P \sqcap Q \sqsubseteq \bot$ is given as:

$$\begin{aligned} & l_{P \sqcap Q \sqsubseteq \bot}(\boldsymbol{c}_P, \boldsymbol{c}_Q, r_P, r_Q) \\ =& \max(0, -\|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 + r_P + r_Q + \gamma) \\ &+ \big| \|\boldsymbol{c}_P\|_2 - 1 \big| + \big| \|\boldsymbol{c}_Q\|_2 - 1 \big|. \end{aligned} \tag{2.8}$$

It enforces the inequality $\|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 \geq r_Q + r_P + \gamma$. According to the setting of γ, the user can decide how far the two balls are separated.

2.4 Terminology: Machine Learning

Some commonly used terms during the experiments in this study are presented below.

A **class** in an image classification task is a discrete category that a set of images belong to.

The encoding (also representative of the name or meaning) given to a class is known as a **label**. Sometimes this is referred to as the **ground truth label**.

The set of all classes present in the image dataset is called **label classes**. This is a subset of the classes in the ontology related to a given task in this study.

An **embedding** is a set of numbers representing an object in a continuous vector space. An embedding representing a class is called a **concept embedding** in this study.

A concept embedding representing a label class is called a **label embedding**.

During few-shot image classification, **base learning** is referred to the initial training of the model using classes with many example images. **Fine tuning** is the process that follows base learning, where the prediction layer of the model is further trained using new classes with limited image examples.

Background classes are the classes of images used for base learning step and **few-shot classes** are the classes of images used for the fine-tuning step.

The classes chosen for a given few-shot task are called **candidate classes**. These are a subset of few-shot classes.

Chapter 3

Related work and this Project

The contributions of this thesis fall broadly under the area of hybrid learning systems, where the integration between statistical (also known as data-driven or sub-symbolic) and symbolic (also known as knowledge-driven or semantic) methods is investigated [vBdBvH$^+$21, DRMD$^+$19, APP$^+$20]. This area has seen rapid growth recently [LGG$^+$20, KFR06, DM15], mainly due to the limitations discovered in popular purely data-driven approaches (Section 1.1). Many variations of hybrid learning systems exist that differ according their components such as, input and output types, methods of processing data or symbols and methods of combining information from data and symbols. The applications of these approaches are not limited to vision tasks and I identify their relatedness to the approaches proposed in this study with regard to their hybrid nature.

Inspired by the design pattern formulation from [vBdBvH$^+$21], I categorise and visualise the general flow of each category of approaches in the proceeding sections. In the flow charts, a generalised nomenclature for blocks is followed, where the rectangles represent an instance of *data* that can be *sub-symbolic* (e.g., images, text) or *symbolic* (e.g., labels, axioms, relations), the ovals represent a process such as *infer*, *train* or *embed* that follow either *statistical* or *logic*-based methods and the hexagons represent a *model* which can be either *sub-symbolic* (e.g., neural network) or *symbolic* (e.g., ontology).

3.1 Learning Symbols from Data

One set of approaches found under hybrid learning systems tries to learn ontologies using data in the form of text [AWK$^+$18, Bre08, ESB$^+$18, WLB12]. Intermediate

representations are generated from text via a statistical model and a set of relations forming an ontology is learnt based on them [CMSV09, dBV19]. Similar approaches such as [PMPS18] follow the same flow with a difference in the output, where the aim is to generate a set of class or instance labels. Also approaches such as [BJS17, KC10] start from a set of instance-level relations and learn a class hierarchy based on them. In the area of vision, studies such as [Asa19] investigate on the generation of first-order logic representations from images taken as input. The latter is the most similar to this study in terms of input, but overall these approaches differs in terms of expecting the ontology to become the output of the system. Figure 3.1 shows the generalised flow of the above approaches.

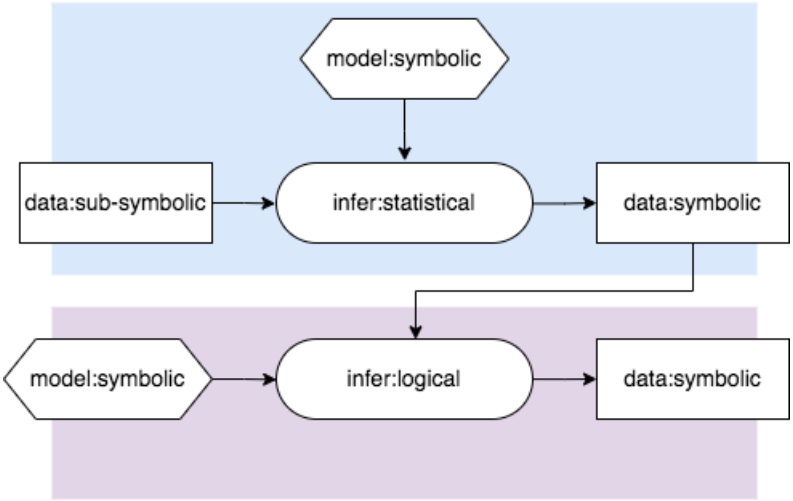


Figure 3.1: Generalised flow of a system that learns symbols from data.

3.2 Using Knowledge to Explain

Another significant set of approaches under hybrid learning systems investigates the generation of explanations for the predictions of statistical models using the knowledge from semantic models [RDGZ+20, XFM+19, RSG16]. These primarily address the 'black-box' issue of the purely data-driven statistical models [Mar18, WB18]. Here, first a machine learning model is trained to predict a label for a given data sample (e.g., an image) in a standard setting [KSH12a]. Then a proceeding semantic model (e.g., knowledge graph) tries to generate a reasoning for the input and output pair (e.g., image and predicted label) according to a chosen knowledge representation. Approaches such as [TdM15] again use knowledge graphs to generate these explanations, whereas

some approaches such as [SXD+17] use a description logic reasoner to generate logic-based reasoning outputs. It is noted that both of these approaches lack the connection to the inner workings of the statistical model when generating the explanations. To this end, approaches such as [CGG+20] use an additional statistical model to model the behaviour of the first and generate explanations in the form of first-order logic formulas.

Figure 3.2 shows the generalised flow of the above approaches. It can be seen that the inference process of the explanation generation section can use only the symbol output or additionally the first model (shown with the dotted line). Although the task of explanation generation is interesting from this set of approaches, they do not explore the opportunity of improving the main statistical model using the knowledge available.

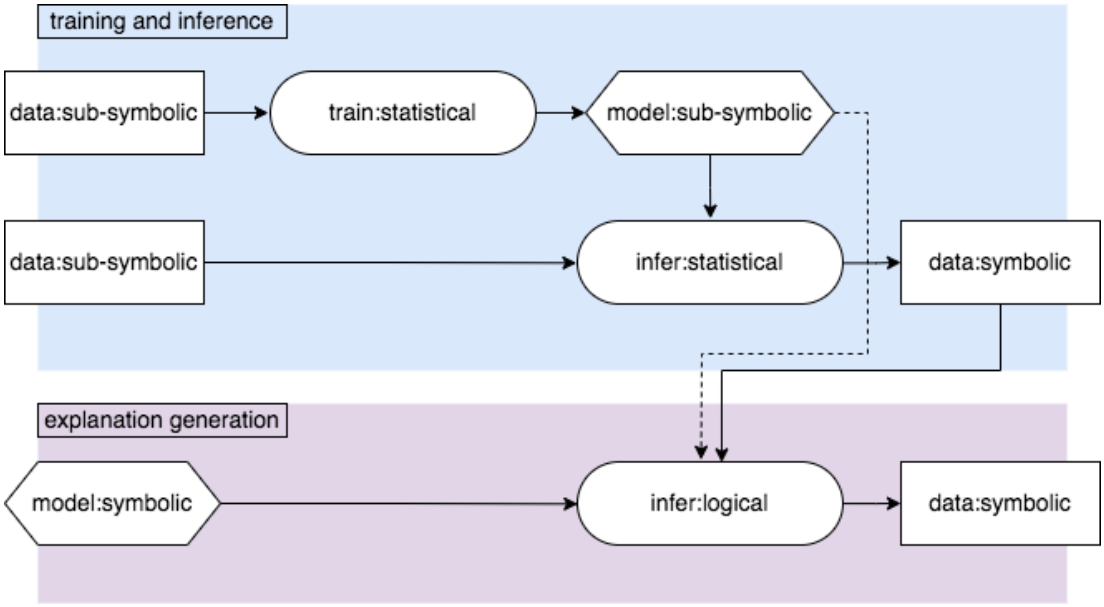


Figure 3.2: Generalised flow of a system that uses a semantic model to generate explanations for the predictions of a statistical model.

3.3 Learning Intermediate Representations

More techniques of hybrid learning can be found where an intermediate symbolic abstraction is learnt and then using it perform another inference procedure. Studies such as [MDK+18] use this flow in the task of performing arithmetic operations on hand written digit images. First the digits are recognised from the images which become the intermediate symbolic representations and then they are added to as the output. The

popular reinforcement learning approach by Deepmind [GAS16] also uses a symbolic representation of the game world during navigation. In further studies, these are found to be usable in downstream reasoning tasks as well [SSS+17].

This intermediate representation can also be in continuous space. Most of the hybrid systems applied in the task of link prediction in knowledge graphs [NMTG15, Pau17], first learn a representation of a graph in a high-dimensional space (also called an embedding) [WMWG17, Ham20]. Next, this embedding is used to infer more knowledge which is ultimately converted back into the symbolic space of the knowledge graph. Figure 3.3 shows a generalisation of the flow of these approaches.

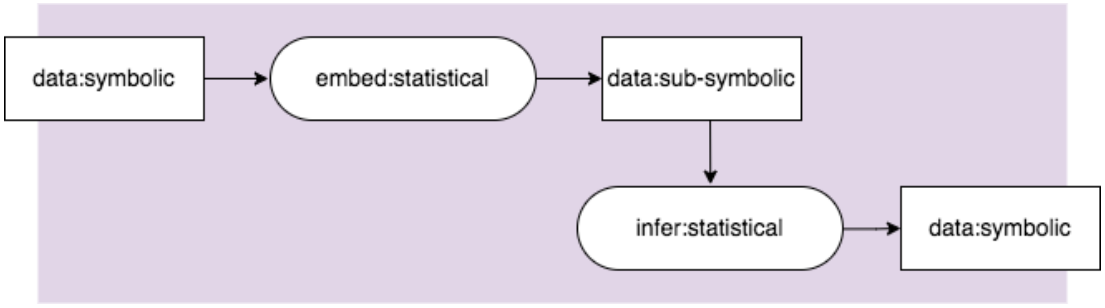


Figure 3.3: Generalised flow of a system that learns intermediate representations.

3.4 Using Logical Rules as Constraints

Another set of techniques provides insights into the use of logical knowledge mainly defined using First Order Logic during the training of neural networks [SDG17, BH19]. Studies such as [TS94] present approaches that use logical rules when deciding the parameters of neural networks but they lack the capability of expanding to deeper networks. With more recent approaches such as [SG16], techniques are presented to guide the learning process of a model to obey the background knowledge provided. Here, logical knowledge act as constraints during training of the statistical model and allows the semantic information to be embedded into the models. The investigation in [SDG17] extends this same approach to perform the task of semantic image interpretation. Other techniques also use Graph Neural Networks [BGLL+20] to embed knowledge from graphs when training neural networks. Figure 3.4 shows the generalised layout of these approaches. I find that these methods can have a higher computational overhead than the approaches proposed in this study, especially when applied for vision tasks.

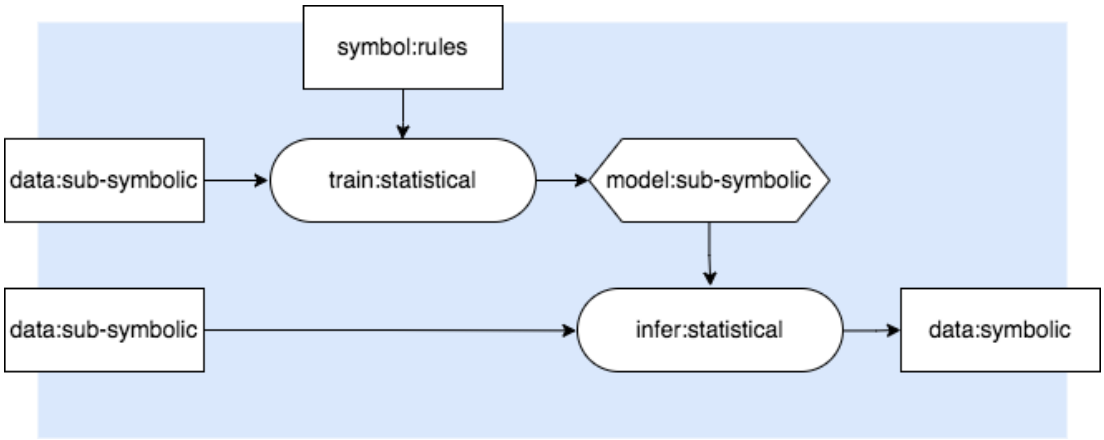


Figure 3.4: Generalised flow of a system that uses logical rules as constraints.

3.5 Capturing Reasoning Capabilities

A key feature of symbolic approaches has been the ability to reason about the given knowledge to infer new knowledge [Pan06]. Statistical methods lack this ability [HL20] and several studies have specifically looked into capturing the reasoning capabilities of symbolic methods in statistical methods. Approaches such as [HL17] show the ability of a neural network to learn about entailments such as membership of an individual instance to a class and existence of relations. Another study [ESB+18] shows how the RDF knowledge graph reasoning can be learnt via embedded triples. Generally with these approaches, the results of a reasoning task become the data for the statistical model as shown in Figure 3.5.

3.6 Informed Machine Learning

I identify the approaches under this section as the most similar to the proposed approaches in this study. Under informed machine learning, the training stage of a statistical model is guided by the knowledge obtained from a related symbolic model [VRMG+19]. Sometimes this used knowledge is also called a symbolic prior to the whole system [BGL14].

One of the main common properties of these approaches is the formulation of a semantic loss function for the statistical model that is influenced by the background knowledge [XZF+18, DLAG+20]. In approaches such as [DLM20, MDG+20], the goal of the loss function is to satisfy the constraints provided by knowledge. In [DRR16], logical rules are used during the gradient decent learning phase. Approaches

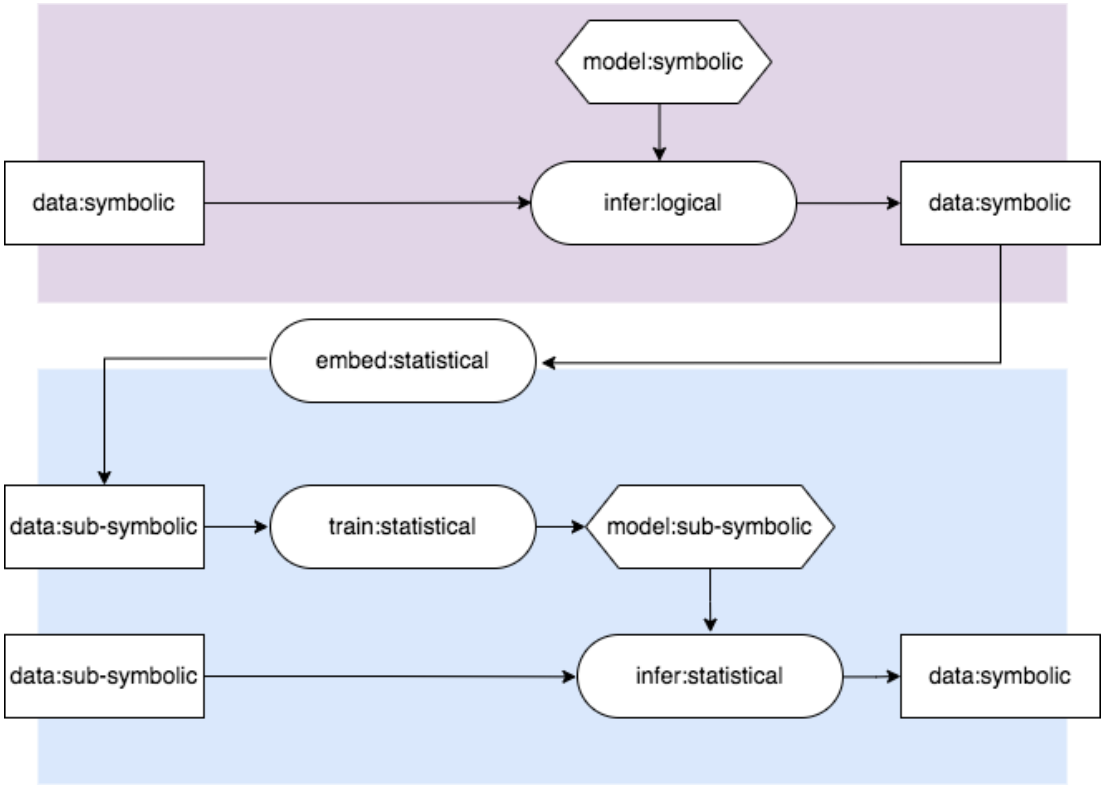


Figure 3.5: Generalised flow for a statistical model to learn reasoning from a symbolic method.

such as [SDSGR18] investigate similar techniques in hyperbolic spaces. The previously discussed [SG16] can also relate to this category of systems, where the minimisation of the neural network loss is driven by the satisfiability with the background knowledge in FOL form. Similar techniques can be seen in reinforcement learning as well [Gei06, IYIM20], where the agent's behaviour is governed by constraints in the symbolic space. Several more studies have further shown the effectiveness of using prior knowledge as constraints when guiding the learning system [SLM21, MGDG19]. The use of knowledge graphs and ontologies as the source of this background knowledge is also seen [KHTT$^+$16, BMT17]. Another variation can be seen in approaches such as [ABRS20, ZPW$^+$19, FZD$^+$19, FSCO18], where the symbolic knowledge becomes the output and is fed back to the learning model in an iterative process. Overall, all these approaches inspire the thinking behind the investigations in this study. Figure 3.6 shows a generalised structure of the approaches in informed machine learning. The difference between this and the flow in Figures 3.5 is the use of the symbolic information during the training process rather than as input data.

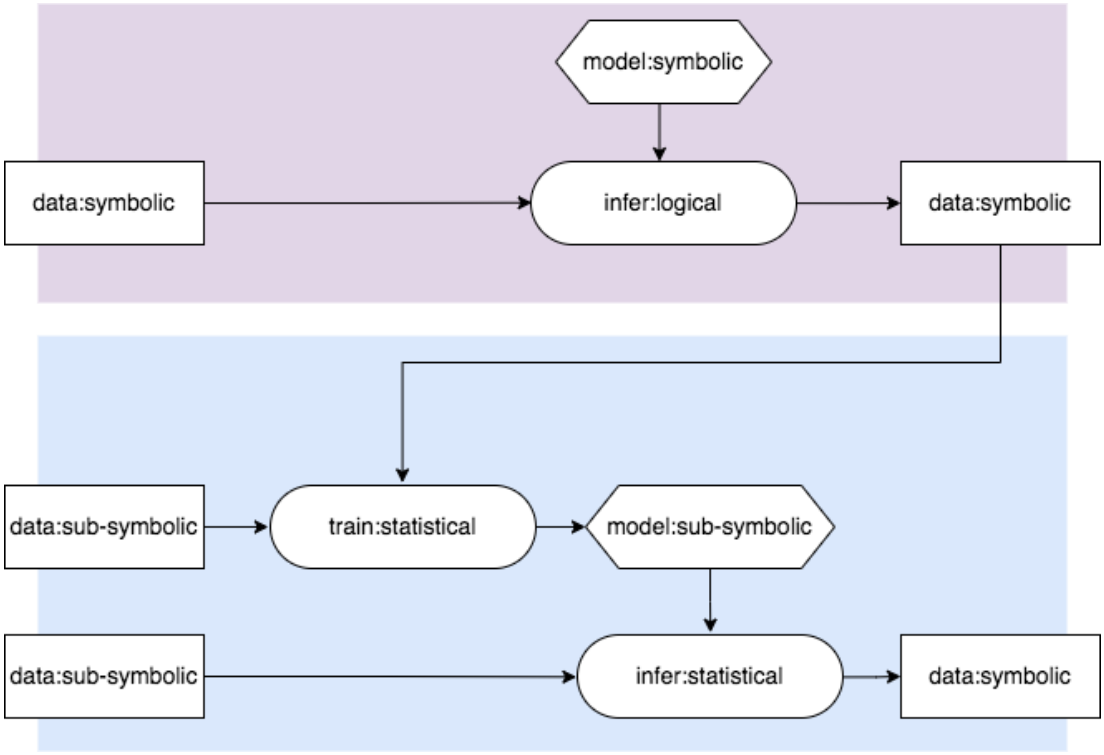


Figure 3.6: Generalised flow of a hybrid system performing informed machine learning.

3.6.1 Further Categorisation of Informed Machine Learning

Since informed machine learning techniques are closely related to this thesis, I further look into the different nuances of these systems. Rueden et al. [VRMG+19] proposes an effective framework when categorising these approaches according to three criteria; (1) the type of background knowledge used, (2) the transformation done on that knowledge and (3) the stage at which the knowledge is integrated with the statistical model. Figure 3.7 shows this framework with possible choices for each criteria. The existing work under informed machine learning can be categorised according to the choices they make.

According to [VRMG+19], the knowledge types are decided qualitatively, looking at whether they are formalised or not formalised knowledge [FPSS96]. Background knowledge used in a given system can also be a combination of these multiple types. In this study, I focus on formalised knowledge in terms of an ontology. When considering the criterion relating to the transformation of knowledge, [VRMG+19] presents choices ranging from rules to human interaction. In the case of this study, I identify the choice of constraints as the closest idea relating to the approaches proposed in this

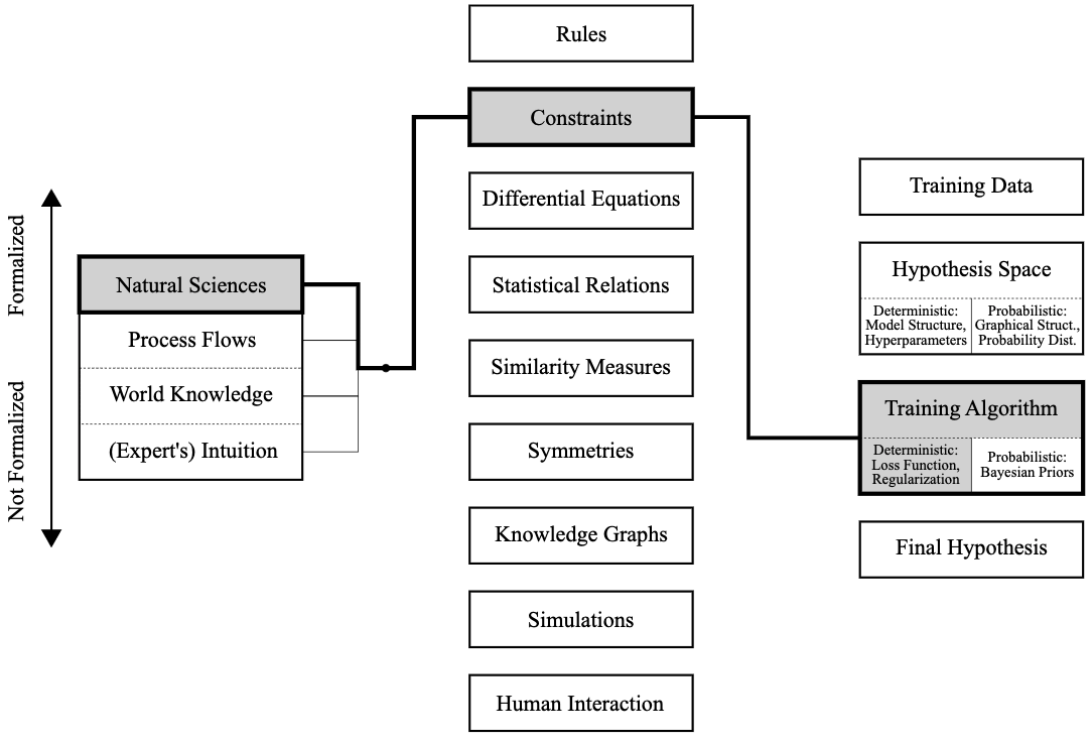


Figure 3.7: The choices for each criteria when categorising informed machine learning approaches. The knowledge types range from formalised to not formalised representations. A given approach can be found to take a path along these choices. I identify the highlighted path as the most similar to the approaches proposed in this study; taken from [VRMG+19]

study, although it is not fully representative of the concept embedding learning process.

Regarding the third criterion of knowledge integration, this study focuses on modifying the loss functions during the training process to incorporate the background knowledge. This is similar to approaches such as [XZF+18, MCX18] where minimising the guided loss function integrates the knowledge from the symbolic source to the neural networks. Also approaches such as [HML+16, DGS17, DRG17, DGS16, ELBS+15] follow the same integration point but with knowledge as logical rules. In terms of informing the loss functions with embedded symbolic knowledge, the approaches of this study is more related to work such as [WYG18b, FCS+13, GCC+21], although they differ when it comes to the sources of knowledge and the downstream vision task used for evaluation. Apart from this, the existing work presents the possibilities of knowledge integration in the training data as additional features or labels [KWRK17, RSS+18, PZL+18, WWX17, LGF16b], in the hypothesis space that determines the structure of the models and hyperparameters [RD06, KBB+12, JZSS16,

N+04, BPL+16, JXLL18b] and also in the final hypothesis phase where the outputs of the algorithm is assessed according to the background knowledge [CFN16].

I identify the integration of background knowledge at the loss function during the training of informed machine learning approaches as a strong point. This allows the prediction space to be defined by the representations obtained via the background knowledge, enhancing the interpretability of the predictions.

But the learning of representations from background knowledge can come with a loss of information. This is a challenge when using an informed machine learning approach and this thesis attempts to address it with the concept embeddings techniques proposed.

3.7 Research Questions for this Project

Looking at the related work, I identify several gaps in the understanding about the application of informed machine learning approaches when performing vision tasks. Based on these, I formulate the research questions to be answered in this project as follows.

3.7.1 Can an ontology be a suitable source of background knowledge for an informed machine learning approach performing few-shot image classification?

Multiple studies [NMB+13, FCS+13] show how text-based knowledge can be used when informing a deep learning model. Approaches such as [WYG18b] show the effectiveness of the use of symbolic information from knowledge graphs during a vision task. But ontologies can provide richer and more consistent knowledge along with additionally inferred information compared to text or knowledge graphs. Also, large ontologies already can be found with expert knowledge about specific domains [WNS+11, Leo08] that can be used as sources of background knowledge. Hence, I argue that investigating how to make use of ontologies in informed machine learning approaches are important. Further the research question focuses on the vision task of few-shot image classification with this regard.

3.7.2 Is learning ontology-based concept embeddings and using them to guide the loss function during the training of a deep convolutional neural network a good strategy when integrating background knowledge?

Another point is addressing the method of ontology-based knowledge integration during the training phase of a neural network. Approaches such as [SDG17] use ontology-based knowledge in the form of constraints that should be satisfied by the learning process. Approaches such as [FCS+13] show that guiding a neural network training process with external embeddings is effective, although there is a lack of understanding as to how a similar method would work with embeddings generated from an ontology. Hence, addressing the technique in which ontology-based knowledge can be integrated is important. Is embedding the knowledge in a continuous vector space a reasonable approach? Given embeddings, how to effectively use them when guiding the training of a DCNN? I identify these question as important to address.

3.7.3 What type of concept embeddings from ontologies are well suited for the aforementioned task?

Next, looking at the generation of ontology-based embeddings, it is important to understand what methods can be used to learn a faithful representation of an ontology in a vector space. A faithful representation would closely relate similar concepts with each other in the space. The relatedness is governed by the distance or the similarity between concepts.

I see a drawback in approaches such as [CHJR+21, HMCJR19a], where an intermediate step converts a portion of the ontology into a graph structure. The design space when converting ontology-based information into graphs can be very large, meaning an ontology can give rise to many types of graphs. When embedding these graphs, there can be a loss of information. Whereas approaches such as [KLWYH19] can learn embeddings directly using knowledge in the form of axioms. Since axioms can be classified using a reasoner, it can be ensured that equivalent ontologies produce similar sets of embeddings. Furthermore, more geometrical features in the vector space are exploited when representing the properties in the ontology. Also, the resulting embeddings are not highly dependent on the ontology syntax in these approaches.

There exist a gap in the understanding of the impact of these different embedding

learning techniques when using them as a part of an informed machine learning system. How faithfully each embedding type represent an ontology? What geometrical features of the embeddings help the downstream task? This study aims to address these questions.

3.7.4 What type of ontology information will contribute towards good concept embeddings when helping a few-shot image classification task?

An ontology can capture several types of relationships among a set of concepts such as their hierarchical structure with subclass relations and additional features about each concept with class expressions containing object properties [MPSP$^+$09b] (e.g., Dog *SubclassOf* Animal and Dog *hasPart* some Tail). During the embedding of this knowledge, concept embeddings can capture either all or a selection of these relationships. So, it is important to understand what choices are helpful given a few-shot image classification task. For example, when classifying bird images, would information such as the colour or shape of a bird be more useful than the hierarchical structure of the different breeds? I identify that addressing this point for a visual task is important.

3.7.5 How to evaluate the errors during image classification with respect to the background knowledge used in an informed machine learning approach?

I identify an extension to the existing evaluation methods for image classification performance, in the present of background knowledge. Currently, an error during any image classification task is captured if the the predicted class for an image does not exactly match its ground truth label [KSH12a, HQDN19]. But in an informed machine learning approach, I argue that the degree of an error can be evaluated based on the background knowledge used in the task. For example, classifying an image of the class *Poodle* as a *Dog* is a smaller error than classifying it as a *Fish*. This idea is addressed in this study during the evaluation of few-shot image classification results of informed machine learning approaches.

Chapter 4

Ontology-based Concept Embeddings

4.1 Ontology Construction with Image Datasets

To investigate the research questions of this study, suitable OWL ontologies were required that contained background knowledge about the class labels of relevant image datasets. These were not available from existing studies, hence in this chapter I describe how new ontologies were constructed using knowledge sources and dataset annotations.

Throughout the study, I make use of four different image datasets to experiment with few-shot image classification, namely, miniImageNet [VBL+16a], tieredImageNet [RTR+18], Stanford Dogs dataset [KJYFF11] and Caltech-UCSD Birds-200-2011 [WBW+11a] (see Sections 4.2.1 and 4.3.1). The class labels of the first three are based on WordNet [Mil95] which was used to obtain information about the class hierarchy when constructing their ontologies. The latter dataset was chosen to investigate multi-relational knowledge in ontologies, where I use the attribute annotations provided with the dataset to construct the ontology. I explain these two techniques separately in the following sections.

4.2 From WordNet to Ontology

4.2.1 Datasets

- **miniImageNet** [VBL+16a] consists of 60,000 images, each labelled with one of 100 classes from ImageNet [DDS+09b]. Each class contains 600 example

images. The classes cover a variety of objects from animals such as *miniature_poodle* to music instruments such as *oboe*.

- **tieredImageNet** [RTR+18] is larger than miniImageNet, containing 608 classes from ImageNet. Its classes are chosen according to on 34 human-curated higher-level categories. They are subdivided accordingly to ensure more distinction among the training and testing stages. For example, classes related to the higher-level category *musical_instruments* are not split between training and testing sets.

- **Stanford Dogs** [KJYFF11] is a fine-grained classification dataset composed of 120 dog species from ImageNet. A total number of 20,580 images span among the classes. This dataset offers a more challenging task as similar-looking classes can be harder to classify especially when learning with few examples.

4.2.2 Ontology Construction

The above datasets are subsets of ImageNet [DDS+09a], hence their class labelling is based on synsets of WordNet [Mil95].

I chose the hypernym tree of WordNet to be the source of knowledge about the label class hierarchy in this study. Given a label, the corresponding synset name (e.g., *miniature_poodle.n.01*) together with all more general classes until *entity.n.01* (top entity in WordNet) were extracted. When constructing the ontology, the hierarchy according to the hypernym tree formed the subsumption relationships among the classes (e.g., *miniature_poodle.n.01 SubClassOf dog.n.01*). All sibling classes that lie in the same hierarchical level were defined to be disjoint (e.g., *miniature_poodle.n.01 disjointWith german_shepherd.n.01*).

Figure 4.1 shows a snapshot of the class hierarchy of the ontology constructed for the miniImageNet dataset. It can be observed how a class label such as *miniature_poodle.n.01* is placed in the hierarchy of concepts captured from WordNet. The final statistics of the ontologies of miniImageNet, tieredImageNet and Stanford Dogs are shown in Table 4.1. The number of inferred axioms entailed by each ontology is including all transitive relations entailed by the *SubClassOf* axioms.

Also, I construct two versions of the Stanford Dogs ontology where the 'reduced' ontology removes 12 of the classes at the top of the original ontology to avoid redundancy and leaves "Dog" as the top class.

All ontologies were saved in the OWL Functional Syntax format and classified via

the HermiT OWL reasoner [SMH08] before using in the experiments. These ontologies can be found at `https://github.com/miranthajayatilake/ViOCE-Ontologies`.

Table 4.1: The statistics of the constructed ontologies for miniImageNet, tieredImageNet and Stanford Dogs Dataset.

Ontology	Class count	Number of axioms		Number of inferred axioms
		SubClassOf	DisjointClasses	
miniImageNet	317	316	51	521
tieredImageNet	1096	1095	235	11770
Stanford Dogs	148	147	29	2219
Stanford Dogs (reduced)	136	135	29	521

4.3 From Annotations to Ontology

4.3.1 Dataset

Caltech-UCSD Birds-200-2011 (CUB-200-2011) is an image dataset that includes 11,788 images of 200 bird species. Each image is further annotated with 15 part locations (Figure 4.2(a)), 312 binary attributes and a bounding box localising the bird in the image. The attributes are distributed among 28 groups (Figure 4.2(b)) and are visual in nature pertaining to properties such as colour, shape, pattern, length, or size of a particular body part (e.g., *hasWingColor Black*). A certainty score between 0 and 100 was also provided along with an attribute annotation that represents the confidence of the human that performed the annotation about the presence of each attribute, 100 being the most certain.

4.3.2 Ontology Construction

For the purpose of constructing an ontology of the classes in CUB-200-2011, I make use of the attribute annotations and their certainty scores. Each bird species was defined as a named class in the ontology and all attributes with a certainty score greater than 50 was chosen for each image in a class. All attribute components such as colour, pattern, shape, length, size, and body parts were also defined as concepts to be used in class axioms. As design choices, 'leftLeg' and 'rightLeg' were combined to a concept 'Leg', and similarly 'leftWing' and 'rightWing' was also combined to a concept 'Wing'. I found that the left and right distinction of these part to be trivial due to their visual similarity and lesser importance during image classification. Also, some intermediate concepts were added to group colour and shape properties such as 'Blue-ish',

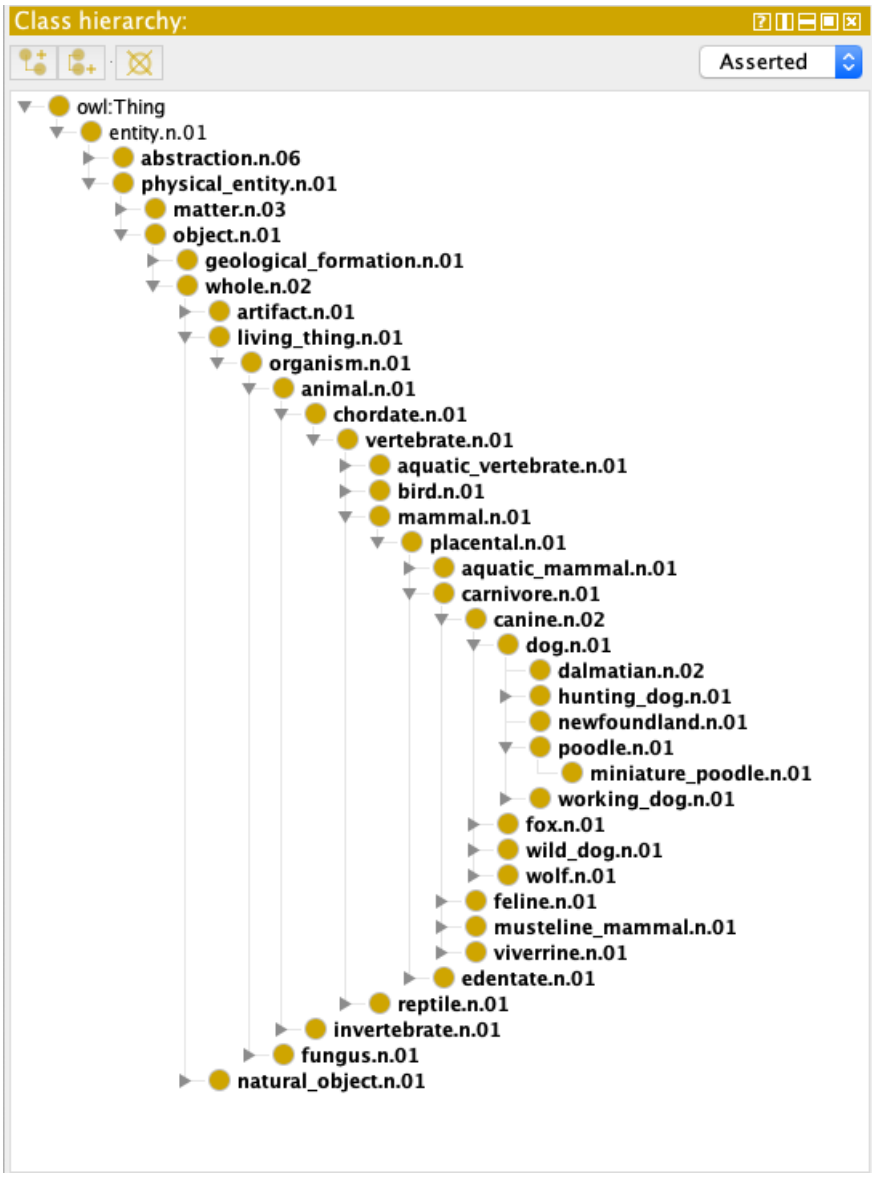
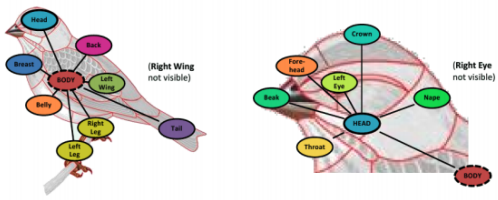


Figure 4.1: A snapshot of the class hierarchy of miniImageNet ontology constructed using OWL

'Brown-ish', 'BillShape' and 'BodyShape' etc. Body parts were also grouped under more general concepts such as 'Head' and 'Body'. These were added to make the knowledge about the different properties more explicit. All sibling classes were made disjoint. I name the resulting ontology as 'Birds Ontology' in this study.

Two variations of the Birds Ontology were created to be used in the methods presented in Section 4.8 as detailed below. These ontologies can be accessed via `https://github.com/miranthajayatilake/CUB-200-2011-OWL-Ontology`.



(a) Collected Parts

Part	Attributes	Part	Attributes	Part	Attributes
Beak	*HasBillShape, HasBillColor , HasBillLength*	**Back**	*HasBackColor, HasBackPattern*	**Breast**	*HasBreastPattern, HasBreastColor*
Belly	*HasBellyPattern, HasBellyColor*	**Fore-head**	*HasForehead Color*	**Bird** (all parts)	*HasSize, HasShape*
Throat	*HasThroatColor*	**Nape**	*HasNapeColor*	**Head**	*HasHeadPattern*
Crown	*HasCrownColor*	**Eye**	*HasEyeColor*	**Leg**	*HasLegColor*
Tail	*HasUpperTailColor, HasUnderTailColor, HasTailPattern, HasTailShape*	**Wing**	*HasWingPattern, Has WingColor, HasWingShape*	**Body**	*HasUnderpartsColor, HasUpperPartsColor, HasPrimaryColor*

(b) Attribute Part Associations

Figure 4.2: CUB-200-2011 annotated parts and attributes for each image; taken from [WBW+11a] (a) 15 part locations. (b) 28 attribute-groupings.

1. Birds Ontology with *implicit* body parts (BO-IBP)

   In this ontology, I define the class axioms of the 200 bird spices with object properties that *implicitly* contain information on body parts as shown in 4.1, i.e., this ontology does not use class names for body parts such as *Wing* but property names such as *hasWingColor*. This is directly using the attribute naming format from the annotations in CUB-200-2011.

$$\begin{aligned} &American\_Crow\ SubClassOf\ Bird\ and \\ &\qquad (hasWingColor\ some\ Black)\ and \\ &\qquad (hasWingPattern\ some\ Solid) \end{aligned} \tag{4.1}$$

2. Birds Ontology with *explicit* body parts (BO-EBP)

   The class axioms of the 200 bird concepts in this ontology *explicitly* contain the body part concepts together with the new object property *hasPart* as shown in axiom 4.2. New object properties were defined such as ‘hasColor’, ‘hasPattern’, ‘hasShape’ ‘hasLength’ and ‘hasSize’ to assert the respective features. I produce this variation of the Birds Ontology to match the requirements of the embedding methods in Section 4.8.3 and 4.8.2.

$$\begin{aligned} &American\_Crow\ SubClassOf Bird\ and \\ &\qquad (hasPart\ some\ (Wing\ and\ hasColor\ some\ Black))\ and \\ &\qquad (hasPart\ some (Wing\ and\ hasPattern\ some\ Solid)) \end{aligned} \tag{4.2}$$

Details about the above ontology variations are shown in Table 4.2. The reason for the reduction of the number of object properties in BO-EBP, compared to BO-IBP, is due to the modification explained in (2) above.

Table 4.2: Variations of the Birds Ontology

Ontology	Number of labelled classes	Number of object properties
BO-IBP	285	26
BO-EBP	303	6

4.4 Ontology-based Concept Embeddings

In this Section, I investigate the encoding of ontology-based symbolic knowledge into continuous vector spaces by learning concept embeddings. Two methods are proposed to achieve this task, one capturing the similarity between concepts in an ontology to define a point in space for a given concept and the other directly embedding axioms as n-balls in space. Experiments were carried out using the three ontologies constructed with classes of the three image datasets that are used as benchmarks in few-shot image classification. I discuss the quality of the resulting embeddings in terms of their ability to represent the knowledge from the input ontologies.

4.5 Why Ontology Embeddings?

In Section 1.3.5, I argue that an ontology is a richer and a reliable source of knowledge when compared to others such as text or knowledge graphs. The properties of an ontology such as knowledge inference and assurance of consistency qualify them to be an ideal candidate to be used in the integration of background knowledge with vision models.

In terms of what knowledge can be represented by an ontology, I identify two main types that can be beneficial for the downstream image classification tasks. First is hierarchical knowledge about the classes in a given task. For example, it needs to be investigates whether informing a model with knowledge such as classes ‘Poodle’ and ‘German shepherd’ belong to a more general class ‘Dog’ would help the downstream task. The other type of knowledge that can be important is relations of a class. For example, I investigate whether it would be helpful to inform a vision model about the

'colour' or 'shape' of a 'Bird' during a classification among breeds of birds. Ontologies can capture both of these knowledge type efficiently via axioms.

Considering the exiting work on the integration of ontologies with broader machine learning approaches (not limiting to vision tasks) [VRMG+19], three methods can be identified considering the form in which ontology-based knowledge is used. First is the extraction of graphs from ontologies and using them either to extend the input space or inform the loss function [GCC+21, RSR15]. Secondly, several approaches can be found to use ontologies as rules providing constraints during the learning process [SG16]. Third is the embedding of knowledge from ontologies as vectors that are then used as additional input [CHJR+21] or guidance to the loss function [FCS+13]. Machine learning approaches primarily deal with operations in continuous vector spaces and motivated by the success of methods such as metric learning [KZS15], I choose the third option of embedding ontology-based knowledge and guiding the behaviour of the prediction space of a DCNN using these embeddings. Also, I find that this approach increases the chances of understanding the behaviour of a machine learning model when embedding images which can potentially contribute towards better explainability of predictions in future work [JMS21a].

When it comes to embedding ontology-based knowledge, I further distinguish the methods that directly capture axioms [KLWYH19] from methods that involve an intermediate step of extracting knowledge graphs from ontologies [CHJR+21]. I argue that the latter suffers from higher loss of information since an ontology can give rise to number of different graphs, e.g., transitive reduct and transitive closure. Hence, a single graph will not be able to represent all knowledge in an ontology. Therefore, in this study I focus on computing an embedding for each concept in a given ontology directly using its axioms. A good ontology embedding should represent all concepts $\widetilde{O}_C$ of an ontology O in a meaningful manner that reflect the relationships between them.

4.6 Concept Similarity-based (*CSim*) Embeddings

Existing methods of embedding text and knowledge graphs [MCCD13b, HYL17] show that a good embedding should project similar meaning words or highly related entities to points in a vector space that are in close proximity to each other. For example, in a good word embedding, the euclidean distance between two points representing the words *Dog* and *Poodle* will be less than that of *Dog* and *Car*. Word embeddings capture these similarities via the co-occurrences of words in text [MCCD13b], while graph

representation leaning methods capture similarities between entities via the number of edges between them [Ham20]. This motivated me to investigate on a similar technique for embedding an ontology, where similarities between concepts can be captured via their subsumption axioms.

Concept similarity-based (*CSim*) embeddings aim to map all named concepts $\widetilde{O}_C$ of an ontology O as points in a vector space that would best represent the similarities among $\widetilde{O}_C$ according to the structure of the class hierarchy of O. Given O as input, a square matrix $\boldsymbol{M}$ of dimensionality $|\widetilde{O}_C|^2$ is generated where each element $\boldsymbol{M}_{i,j}$ in i^{th} row and j^{th} column represents the similarity score between concepts P_i and P_j ($i, j = [1, |\widetilde{O}_C|]$). I choose the atomic similarity between concepts introduced in Section 2.2.2.1 to compute these scores.

Next, dimensionality reduction is applied on $\boldsymbol{M}$ using principal component analysis (PCA) (Section 2.3.2.1) to get the desired dimensionality for the embeddings. I keep this step as optional to use when $|\widetilde{O}_C|$ is too large. From the resulting matrix, a row M_i is chosen as the embedding for concept P_i.

4.7 n-ball Embeddings

I find that the *CSim* embeddings in Section 4.6 offer some ambiguity when representing the exact structure of the class hierarchy of an ontology. For example, relationships such as $Poodle \sqsubseteq Dog$ and $Dog \sqsubseteq Animal$ are both encoded by the proximity of the vectors representing $Poodle$, Dog and $Animal$ in the embedding space. But this arrangement does not make it clear that $Animal$ is a more general class than Dog and $Poodle$ is a more specific class than Dog. This drawback of point embeddings motivated me to look into richer geometric representations in the embeddings space.

An n-ball can represent a concept in the embedding space using both its centre and radius features (Section 2.3.3). This enables the representation of ontology information such as subsumptions and disjointnesses using the positioning of the n-balls in space, e.g., $P \sqsubseteq Q$ will be represented by the n-ball of Q enclosing that of P and $P \sqcap Q \sqsubseteq \bot$ represented by the n-balls of P and Q being seperated from each other.

I build upon the EL embedding technique [KLWYH19] to learn a set of n-balls for all concepts $\widetilde{O}_C$ in the ontology O, which are referred to as *concept embeddings*. I use subsumption and disjointness axioms from the class hierarchy of the ontology O. I use the inferred class hierarchy (*ICH*) (Section 2.1.1) that includes all inferred subclass relations (e.g., if *Poodle SubclassOf Dog* and *Dog SubclassOf Animal* are asserted,

ICH includes *Poodle SubclassOf Animal*). I introduce a regularisation term in (4.4) to prevent radius shrinkage. Following Eqs. (2.7) and (2.8), the radius of the learned n-ball for a leaf concept, which corresponds to a label class, can end up being very small, in order to fit into the balls of its subsumer concepts. Since in the image embedding learning, I will try to map each image to a point inside the n-ball corresponding to its ground truth class, an overly small radius can affect the learning accuracy. Also, to improve the embedding quality, I introduce extra hyperparameters $\psi, \phi > 0$ in (4.3) to explore potentially more expressive design spaces, which is supported by an additional parameter tuning process. This was done to control the embedding quality as the class hierarchy of the input ontology becomes larger. Finally, I minimise the following loss function:

$$l_c\left(\{\boldsymbol{c}_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P, r_Q\}_{P,Q\in\widetilde{O}_C}\right) \tag{4.3}$$

$$= \sum_{P\sqsubseteq Q\in ICH\ (O)} \max(0, \|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 + r_P - r_Q - \gamma)$$

$$+ \sum_{P\sqcap Q\sqsubseteq\bot\in O} \max(0, -\|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 + r_P + r_Q + \gamma)$$

$$+ \sum_{P\in\widetilde{O}_C} \max(0, \psi\sqrt{N_h - L(P)} - r_P) \tag{4.4}$$

$$+ \sum_{P\in\widetilde{O}_C} N(P)\big|\|\boldsymbol{c}_P\|_2 - \phi\big|. \tag{4.5}$$

Here, N_h denotes the total level number contained in the inferred class hierarchy, and $L(P)$ denotes the level of the concept P in this hierarchy, e.g., the top-most concept has level 1. $N(P)$ denotes the number of times the concept P appears in the extracted axioms. Eq. (4.4) restricts the radius of the concept P 's n-ball to be no less than $\psi\sqrt{N_h - L(P)}$. The top-level concepts are allowed to have larger n-balls than the bottom ones.

4.7.1 Embedding Quality and Hyperparameter Tuning of n-ball Embeddings

The n-ball concept embeddings are learnt by minimising Eq. (4.3), e.g., by a gradient descent algorithm. There are three hyperparameters γ, ϕ and ψ to be set. I perform hyperparameter tuning by examining how much knowledge entailed by the ontology is captured by the learnt embeddings.

Three parameter tuning scores are proposed by examining whether $\|\boldsymbol{c}_P - \boldsymbol{c}_Q\| \leq r_Q - r_P$ holds for a ground truth subsumption $P \sqsubseteq Q \in ICH\,(O)$. All the ground truth subsumptions are considered as positive instances. If the inequality holds, it is considered as a positive prediction. The classical F_1 score, which is the harmonic mean of the precision and recall, is used to assess the prediction accuracy of these subsumptions. I calculate two versions of F_1 score, one is referred to as $F_1^{(\text{all})}$ based on all the subsumptions in $ICH\,(O)$. The other only considers the subsumptions involving the leaf concepts, which correspond to all the classes in C_B and C_F, as well as their direct parent classes. This score is referred to as $F_1^{(\text{leaf})}$.

The third parameter tuning score $R^{(dis)}$ examines the disjointness between the leaf concepts. Enumerating all pairs of leaf concepts[1], $R^{(dis)}$ computes the recall of disjointness axioms in the embedding space using the condition $\|\boldsymbol{c}_P - \boldsymbol{c}_Q\| > r_P + r_Q$. A higher $R^{(dis)}$ indicates less overlapping between the n-balls of leaf concepts.

I compute these scores as a mandatory step at the end of each embedding learning process. Using grid search, the combination of γ, ϕ and ψ is found that results in the highest value for ($F_1^{(\text{all})}$ + $F_1^{(\text{leaf})}$ + $R^{(dis)}$ + BI_{tr}), where BI_{tr} is training accuracy on BI. I include BI_{tr} in this tuning procedure because the objective is to obtain the most favourable set of embeddings for the downstream vision task. The complete concept embedding learning process is elaborated in Algorithm 1.

4.8 Multi-relational n-ball Embeddings

I investigate how to embed knowledge about more properties in addition to subsumption and disjointness between concepts and how they would affect the few-shot image classification performance. For example, will there be an improvement by informing a vision model about body colour and shape of a bird when classifying among different bird species? To answer this question, I propose three methods of embedding additional object properties (e.g., *hasColor, hasShape*) with n-ball embeddings. These methods differ from each other in the way that existential restrictions are handled during the embedding learning process.

Below methods discuss three different approaches proposed to learn n-ball concept embeddings using an ontology with multi-relational data.

[1] In the ontology, I have $P \sqcap Q \sqsubseteq \bot$ for all leaf node P and Q

Algorithm 1: n-ball embeddings learnt for all concepts $\widetilde{O}$ of O

Require: Input O
  Best overall score $S^B = 0$
  List of choices for ϕ, ψ and γ in Eq. 4.3
  Best combination of ϕ, ψ and γ ($B_{\phi,\psi,\gamma}$)
  for all possible combinations of ϕ, ψ and γ **do**
    for all $P \sqsubseteq Q \in ICH\,(O)$ and $P \sqcap Q \sqsubseteq \bot \in ICH\,(O)$ **do**
      $l_c\left(\{\boldsymbol{c}_P\}_{P\in\widetilde{O}}, \{r_P\}_{P\in\widetilde{O}}\right)$ (Eq. 4.3)
    end for
    Compute $F_1^{(\text{all})}$, $F_1^{(\text{leaf})}$ and $R^{(dis)}$
    Compute $S = F_1^{(\text{all})} + F_1^{(\text{leaf})} + R^{(dis)} + BI_{tr}$
    if $S > S^B$ **then**
      $S^B = S$
      $B_{\phi,\psi,\gamma} = \phi, \psi, \gamma$
    end if
  end for
  return $B_{\phi,\psi,\gamma}$, S^B

4.8.1 Flattened Ontology Embedding Method (FO-EM)

According to the previous results in Section 4.10, I discover that the proposed n-ball embedding learning method in Section 4.7 is capable of accurately representing the concepts of an ontology bound by the relations of subsumption and disjointness. Inspired by this, the Flattened Ontology Embedding Method (FO-EM) tries to fit the challenge of embedding additional object properties with the same proven method. This is done by 'flattening' the input ontology by defining each attribute of a class as a new named class, reducing all relations to subsumptions.

For example, taking a class axiom such as,

$$\begin{aligned} &\mathit{Anna\_Hummingbird\ SubClassOf\ Bird} \\ &\qquad \mathit{and\ (hasBillColor\ some\ Black)} \\ &\qquad \mathit{and\ (hasBillShape\ some\ Needle)}, \end{aligned} \tag{4.6}$$

if the term *(hasBillColor some Black)* is collectively represented by a new class C_1 and *(hasBillShape some Black)* by C_2, then it can be written as,

$$\mathit{Anna\_Hummingbird\ SubClassOf\ Bird\ and}\ C_1\ \mathit{and}\ C_2. \tag{4.7}$$

Axiom 4.7 implies that,

$$\begin{aligned} &\mathit{Anna\_Hummingbird\ SubClassOf\ Bird} \\ &\mathit{Anna\_Hummingbird\ SubClassOf\ } C_1 \\ &\mathit{Anna\_Hummingbird\ SubClassOf\ } C_2. \end{aligned} \tag{4.8}$$

Axiom 4.8 can now be used to learn embeddings with the same method proposed in Section 4.7. When doing so, I also define that $\mathit{Bird} \sqcap C_1 \sqcap C_2 = \bot$. Conceptually, C_1 is a class that holds things that have a black bill colour and class C_2 holds things that have a needle-like bill shape. *Anna_Hummingbird* is a *Bird* that is also a subclass of C_1 and C_2. Hence, in the resulting embeddings, the n-ball of *Anna_Hummingbird* should be enclosed by the n-balls of *Bird*, C_1 and C_2, while *Bird*, C_1 and C_2 do not overlap.

The loss function for FO-EM is a modification of Equation 4.3, with a new class definition function y added to convert $\exists E.Q$ into a concept, where E denotes an object property and $\boldsymbol{c}_{y(\exists E.Q)}$ and $r_{y(\exists E.Q)}$ denote the centre and radius of the n-ball representing the concept $y(\exists E.Q)$. Equation 4.9 shows the full loss function with the added component 4.10.

$$l_{FO-EM}\left(\{\boldsymbol{c}_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P, r_Q\}_{P,Q\in\widetilde{O}_C}, \{\boldsymbol{c}_{y(\exists E.Q)}\}_{y(\exists E.Q)\in\widetilde{O}_C}, \{r_{y(\exists E.Q)}\}_{y(\exists E.Q)\in\widetilde{O}_C}\right) \tag{4.9}$$

$$\begin{aligned} &= l_c\left(\{\boldsymbol{c}_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P, r_Q\}_{P,Q\in\widetilde{O}_C}\right) \\ &+ \sum_{P\sqsubseteq y(\exists E.Q)\in \mathrm{ICH}(O)} \max(0, \|\boldsymbol{c}_P - \boldsymbol{c}_{y(\exists E.Q)}\|_2 + r_P - r_{y(\exists E.Q)}) \end{aligned} \tag{4.10}$$

4.8.2 Transformation Embedding Method (TF-EM)

The Transformation Embedding Method takes the most similar approach to [KLWYH19] when embedding multi-relational knowledge. It embeds object properties as points in the embedding space and applies translations to n-balls according to the existential restrictions found in the ontology, e.g., $P \sqsubseteq \exists E.Q$ is represented by the n-ball of P enclosed by the translated n-ball of Q using the embedding e representing E. Complex SubclassOf axioms such as $P \sqsubseteq Q \sqcap Q_2 \sqcap Q_3$, where Q_2, Q_3 are also concepts, should broken down to fit the form $P \sqsubseteq Q$ to be used with this method.

For example, taking a class axiom such as:

$$\begin{aligned} &\mathit{American\_Crow\ SubClassOf\ Bird} \\ &\qquad \mathit{and}\ (\mathit{hasPart\ some}\ (\mathit{Wing\ and\ hasColor\ some\ Black}))\ , \end{aligned} \tag{4.11}$$

I introduce a new concept named *BlackWing* and re-define the axiom as below:

$$\begin{aligned} &\mathit{American\_Crow\ SubClassOf\ Bird} \\ &\quad \mathit{BlackWing\ SubClassOf\ Wing} \\ &\quad \mathit{BlackWing\ SubClassOf\ hasColor\ some\ Black} \\ &\mathit{American\_Crow\ SubClassOf\ hasPart\ some\ BlackWing}. \end{aligned} \tag{4.12}$$

The loss function for TF-EM is defined in Equation 4.13. It again takes inspiration from Equation 4.3 to add radii regularisation and hyperparameter ϕ that governs the allowed canvas area for the embeddings. Term 4.14 governs the translation of n-ball during embedding training. Term 4.15 uses negative samples of existential restrictions that are generated at random by replacing Q in $P \sqsubseteq \exists E.Q \in O$.

$$l_{TM-EM}\left(\{c_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P.r_Q\}_{P,Q\in\widetilde{O}_C}, \{e\}_E\right) \tag{4.13}$$

$$= l_c\left(\{\boldsymbol{c}_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P, r_Q\}_{P,Q\in\widetilde{O}_C}\right)$$

$$+ \sum_{P\sqsubseteq\exists E.Q\in O} \max(0, \|c_P + e - c_Q\|_2 + r_P - r_Q) \tag{4.14}$$

$$+ \sum_{P\sqsubseteq\exists E.Q\notin O} \max(0, -\|c_P + e - c_Q\|_2 + r_P + r_Q) \tag{4.15}$$

4.8.3 Step-wise Partial Embedding Method (SP-EM)

The Step-wise Partial Embedding Method (SP-EM) is an approach taken to learn a set of n-ball embeddings representing concepts with more control over how each n-ball occupies the space. Similar to [KLWYH19], SP-EM learns point embeddings for object properties that are used to translate n-balls according to the existential restrictions found in the ontology. As the initial step, separate sets of concepts are identified according to the class definitions. These concepts are embedded independently and

combined in a 'step-wise' manner until the whole ontology is represented in one embedding space.

The novelty of SP-EM is twofold. First is the use of the intersection of two n-balls representing concepts P and Q translated by E, to represent an axiom $P \sqsubseteq \exists E.Q$, where E is an object property. Second is the geometric construction of a new n-ball representing the collective intermediate concept of $P \sqsubseteq \exists E.Q$. Intersections and constructions are done step by step according to class axioms resulting in 'partial' embeddings. This is a unique feature compared to the other embedding methods proposed in Section 4.8.

Figure 4.3 visualises the behaviour of SP-EM with the class axiom 4.16 as an example. The main sets of concepts identified are colours and body parts that help define the bird class *American_Crow*. So as the first step, n-balls of these concepts are learnt independently using their subsumption and disjointness relations among each other as shown in 4.3 (a). Next, to represent a property $(hasPart\ some\ (Wing\ and\ hasColor\ some\ Black))$, the n-ball representing *Black* is translated using the *hasColor* property and projected to a combined space with the body part n-balls to form the relevant intersections as shown in 4.3 (b). The n-balls for the intermediate concepts such as *Black Wing* are then constructed to occupy the lens at the intersections. Similar procedure is followed with the other two properties as well. Finally, the n-ball of *American_Crow* is learnt to intersect with *Black Wing*, *Black Eye* and *Black Back* as shown in 4.3 (c).

$$\begin{aligned} American\_Crow\ SubClassOf\ Bird & \\ and\ (hasPart\ some\ (Wing\ and\ hasColor\ some\ Black)) & \\ and\ (hasPart\ some(Eye\ and\ hasColor\ some\ Black)) & \\ and\ (hasPart\ some(Back\ and\ hasColor\ some\ Black)) & \end{aligned} \tag{4.16}$$

The loss functions for SP-EM are defined for each step separately. At the initial step of embedding concepts governed by subsumption and disjointness relations, I use the same loss design in Equation 4.3.

The intersection of n-balls for P and Q and the embedding of E when embedding the axiom $P \sqsubseteq \exists E.Q$ is governed by the loss function in Equation 4.17. Here, term 4.18 ensures that the n-ball of P intersects with the n-ball of Q translated by embedding e. Term 4.19 ensures that the n-ball of P is not enclosed by the n-ball of Q translated by embedding e. Furthermore, I include an additional term 4.20 to use the negative samples of existential restrictions that are generated randomly by replacing Q in $P \sqsubseteq$

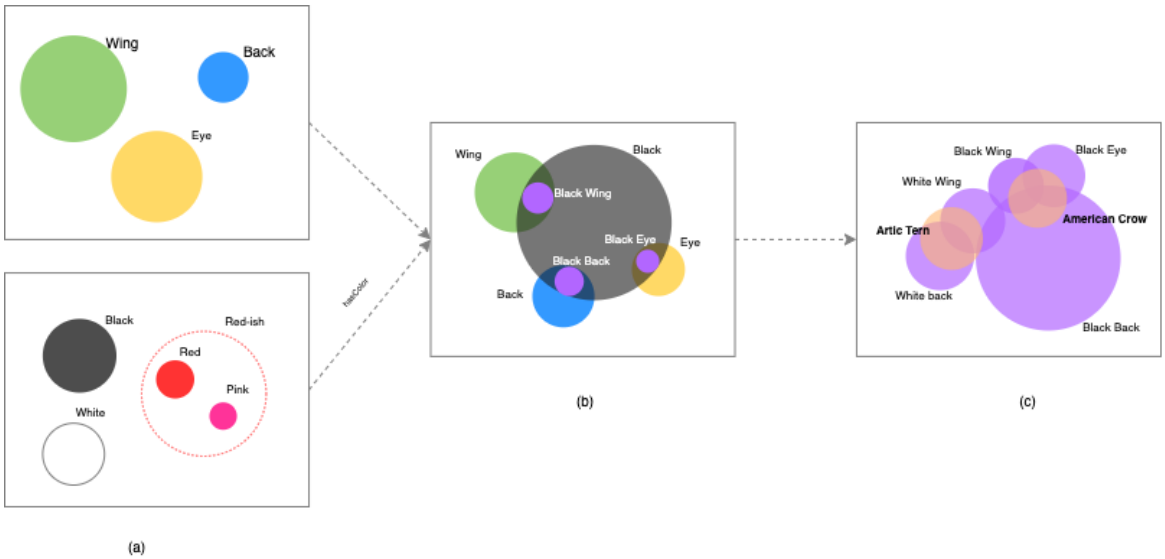


Figure 4.3: An example visualising the behaviour of SP-EM

$\exists E.Q \in O$ during the learning process.

$$l_{SP-EM-1}\left(\{\boldsymbol{c}_P,\boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C},\{r_P,r_Q\}_{P,Q\in\widetilde{O}_C},\{e\}_E\right) \tag{4.17}$$

$$= \sum_{P\sqsubseteq\exists E.Q\in O} \max(0,\|\boldsymbol{c}_P+e-\boldsymbol{c}_Q\|_2-r_P-r_Q) \tag{4.18}$$

$$+ \sum_{P\sqsubseteq\exists E.Q\in O} \max(0,-\|\boldsymbol{c}_P+e-\boldsymbol{c}_Q\|_2-r_P+r_Q) \tag{4.19}$$

$$= \sum_{P\sqsubseteq\exists E.Q\notin O} \max(0,-\|\boldsymbol{c}_P+e-\boldsymbol{c}_Q\|_2+r_P+r_Q) \tag{4.20}$$

Equation 4.21 and 4.22 are used for the geometric construction of an n-ball that fits inside the lens of an intersection of two n-balls, P,Q. They find the centre $\boldsymbol{c}_L$ and radius r_L of the constructed n-ball respectively.

$$\boldsymbol{c}_L = \left(\frac{\boldsymbol{c}_P+\boldsymbol{c}_Q}{2}\right) + \frac{{r_P}^2-{r_Q}^2}{{\|\boldsymbol{c}_P+\boldsymbol{c}_Q\|_2}^2}\left(\frac{\boldsymbol{c}_P-\boldsymbol{c}_Q}{2}\right) \tag{4.21}$$

$${4r_L}^2 = 2({r_P}^2+{r_Q}^2) - \left(\frac{{r_p}^2-{r_Q}^2}{\|\boldsymbol{c}_P+\boldsymbol{c}_Q\|_2}\right)^2 - {\|\boldsymbol{c}_P+\boldsymbol{c}_Q\|_2}^2 \tag{4.22}$$

Finally, the intersection of two n-balls of concepts P and Q is governed by the loss function in Equation 4.23.

$$
\begin{aligned}
l_{SP-EM-2}\left(\{\boldsymbol{c}_P, \boldsymbol{c}_Q\}_{P,Q\in\widetilde{O}_C}, \{r_P, r_Q\}_{P,Q\in\widetilde{O}_C}\right) \\
= \sum_{P\sqcap Q} \max(0, \|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 - r_P - r_Q) \\
+ \sum_{P\sqcap Q} \max(0, -\|\boldsymbol{c}_P - \boldsymbol{c}_Q\|_2 - r_P + r_Q)
\end{aligned} \tag{4.23}
$$

4.8.4 Embedding Quality and Hyperparameter Tuning

I extend the framework presented in Section 4.10 to measure the quality of the resulting embeddings of the three methods FO-EM, SP-EM and TF-EM. In addition to the scores $F_1^{(\text{all})}$, $F_1^{(\text{leaf})}$ and R^{dis}, I introduce two new scores R^{sub} and r^{avg} to address the multi-relational knowledge capturing. R^{sub} measures the recall of all existential restrictions of the complex $SubClassOf$ axioms and r^{avg} measures the average radius of the n-balls representing the label concepts.

Similar to Section 4.10, all the quality measures are used when tuning the hyperparameter ϕ of the embedding loss functions, where the best ϕ should result in the highest value for $(F_1^{(\text{all})} + F_1^{(\text{leaf})} + R^{(dis)} + R^{sub} + r^{avg} + BI_{tr})$. The modified Algorithm 1 including the new scores is shown in Algorithm 2 which is used by all methods, FO-EM, SP-EM and TF-EM.

Algorithm 2: multi-relational n-ball embeddings learnt for all concepts $\widetilde{O}$ of O

Require: Input O
  Best overall score $S^B = 0$
  List of choices for ϕ
  Best ϕ (B_ϕ)
  for all values of ϕ **do**
    Compute embeddings using FO-EM/SP-EM/TF-EM
    Compute $F_1^{(\text{all})}$, $F_1^{(\text{leaf})}$ and $R^{(dis)}$
    Compute $S = F_1^{(\text{all})} + F_1^{(\text{leaf})} + R^{(dis)} + R^{sub} + r^{avg} + BI_{tr}$
    if $S > S^B$ **then**
      $S^B = S$
      $B_\phi = \phi$
    end if
  end for
  return B_ϕ, S^B

4.9 Experiments

4.9.1 Implementation Details of *CSim* Embeddings

CSim embeddings were computed on both miniImageNet and Stanford Dogs ontologies. The total number of label classes in both these datasets was 100, hence the generated matrix $\boldsymbol{M}$ was of dimension 100x100. I do not use dimensionality reduction in this case. The resulting embeddings were of dimension 100.

4.9.2 Implementation Details of *n*-ball Embeddings

I chose the dimensionality of the concept embeddings to be 300 after several trials with values 50, 100, 200, 300 and 500. 300 dimensions produced sufficient expressiveness with an affordable computational cost. All embeddings were initialised within a range [-1,1] at the beginning of the learning process. The learning rate was set to 0.001.

For the hyperparameter tuning process of each ontology, $\phi = \{100, 30, 20, 10, 5\}$, $\psi = \{20, 1\}$ and $\gamma = \{0.01, 0.001, 0.0001\}$. Embedding training was carried out for 100 epochs in all cases.

4.9.3 Implementation Details of Multi-relational *n*-ball Embeddings

Out of the Birds Ontology variations, FO-EM uses the BO-IBP and both TF-EM and SP-EM use BO-EBP. All embedding learning configurations are the same as in Section 4.9.2. During the hyperparameter tuning process, $\phi = \{50, 40, 30, 20, 10, 5\}$

4.10 Discussions on Embedding Quality

4.10.1 Visualising *CSim* Embeddings

I visualise the resulting *CSim* embeddings of the miniImageNet and Stanford Dogs ontologies reduced to 2 dimensions using t-SNE [VdMH08] in Figures 4.4 and 4.5 respectively. As a comparison, I also plot the embeddings generated for the same miniImageNet classes using the existing graph-based WordNet embeddings technique introduced in Section 2.3.2. In all plots, the circles and triangles represent the base classes and few-shot classes respectively, which is a selection of classes required in Section 5.1. The colours represent clusters of similar classes that were found according

to each set of embeddings using k-means clustering with $k = 15$. The labels consist of the relevant WordNet synset id along with the class name.

Comparing Figure 4.4 (a) and (b), it can be seen how *CSim* embeddings produce more distinguishable clusters in the embedding space, such as the noticeable separation is between living things and nonliving things. Also, more granular features such as the similar positioning of dog classes is seen with *CSim* embeddings, whereas the graph-based WordNet embeddings have produced a more uniform distribution without a clear separation of clusters. I argue that the clear separation of similar class groups is a superior capability of the *CSim* embedding learning process.

Figure 4.5 also shows a similar output with *CSim* embeddings produced on the Stanford Dogs ontology, where more similar dog breeds (e,g,. *Walker_hound* and *English_foxhound*) are clearly clustered together in the embeddings space.

4.10.2 Quality Scores of *n*-ball Embeddings

I compare several instances of the resulting *n*-ball embeddings for the three datasets during the hyperparameter tuning process. Table 4.3 shows the $F_1^{(\text{all})}$, $F_1^{(\text{leaf})}$ and S_D scores for a few selected values of ϕ, ψ and γ. The highlighted row for each dataset is the chosen configuration to take forward depending on BI_{tr}. It can be noted that higher ϕ values generally result in higher quality scores but lower BI_{tr} accuracies. Since ϕ controls the overall canvas space that *n*-balls can occupy, higher ϕ values distribute the embeddings further away from each other. This is found to be not favourable for the vision task. The aim of the hyperparameter tuning process is to find the most satisfiable embedding quality that is the most supportive for the vision task.

Figure 4.6 visualises the embeddings for the instances of the miniImageNet ontology. For visualisation purposes, I reduce the dimensionality of the embeddings to 2, hence the smaller *n*-balls appear to overlap. But in 300 dimensions, the majority of these embeddings do not overlap as implied by the S_D score. I further visualise the resulting embeddings for the instances of Stanford Dogs (reduced) ontology in Figure 4.7, where 4.7 (b) shows a set of embeddings learnt before the reduction of the top-most classes. It can be seen that the bigger *n*-balls do not contribute much for the structure of the leaf concepts. This was the reason behind working with the reduced Stanford Dogs ontology.

Table 4.3: Examples of n-ball embedding quality scores for the three datasets during hyperparameter tuning.

Ontology	ϕ	ψ	γ	$F_1^{(\text{all})}$	$F_1^{(\text{leaf})}$	S_D	BI_{tr} (%)
miniImageNet	100	20	0.0001	0.82	0.43	42	54.38
	30	1	0.0001	0.89	0.64	10	67.93
	5	**1**	**0.0001**	**0.86**	**0.54**	**48**	**83.12**
tieredImageNet	5	1	0.0001	0.79	0.35	136	58.13
	20	**1**	**0.0001**	**0.89**	**0.55**	**49**	**78.34**
Stanford Dogs (reduced)	10	1	0.0001	0.87	0.75	30	70.92
	5	**1**	**0.0001**	**0.85**	**0.54**	**45**	**85.84**

4.10.3 Quality Scores of Multi-relational n-ball Embeddings

The tuned values of ϕ with other embedding quality scores for each method during my experiments are shown in Table 4.4. Overall, it can be see that SP-EM produces the highest quality embeddings with respect to all scores. Also it gives the lowest ϕ value with the highest average radius for label concepts, meaning the embeddings take up lesser space compared to the other methods. FO-EM does better in three scores compared to TF-EM. R^{sub} of FO-EM is not computed as it does not embed object properties separately. It can be seen that TF-EM gives the biggest ϕ value, meaning its n-ball embeddings takes up more space when representing the concepts.

Table 4.4: Embedding quality scores and tuned hyperparameter value for FO-EM, TF-EM and SP-EM

Method	$F_1^{(\text{all})}$	$F_1^{(\text{leaf})}$	S_D	R^{dis}	R^{sub}	r^{avg}	ϕ
FO-EM	0.79	0.99	6	0.98	-	1.01	10
TF-EM	0.76	1	0	0.97	0.47	0.67	50
SP-EM	0.96	1	0	0.98	0.53	1.05	5

4.11 Ablation Study

4.11.1 ICH (O) vs Asserted Axioms for n-ball Embeddings

I empirically investigate the effect of the inferred class hierarchy of O during the concept embedding learning process described in Section 4.7. Using the techniques in Section 4.7.1, I measure the quality of embeddings produced with the asserted class hierarchy as input. Here, the embedding procedure has to capture information such as,

if $P \sqsubseteq Q$ and $Q \sqsubseteq E$, then $P \sqsubseteq E$ without an explicit definition. As shown in Figure 4.8*a* using the miniImageNet ontology, the *n*-balls are mostly huddled together with an incorrect interpretation of the class hierarchy when using the asserted hierarchy. The $F_1^{(\text{all})}$ score computed for these embeddings was 0.25, which is quite low. It was observed that with a higher number of classes and levels in the hierarchy (> 50 classes or > 10 levels), capturing transitive knowledge becomes much harder with only the asserted class hierarchy.

By using $ICH\,(O)$ as input to learn embeddings, I provide all the inferred axioms entailed by the ontology. The resulting embeddings in the case of miniImageNet are shown in Figure 4.8 *b*, where the hierarchical structure and the main clusters of concepts are are better defined by the size and placing of the *n*-balls. Here, the $F_1^{(\text{all})}$ score was reported to be 0.88, which is much higher than the previous case. This shows that the inclusion of all inferred axioms as input leads to an effective embedding learning process.

4.12 Summary and Directions

Overall, four new OWL ontologies were constructed to be used in the experiments of this study. For miniImageNet, tieredImageNet and Stanford Dogs ontologies, I use WordNet's hypernym tree as the knowledge source to construct the class hierarchies. I concentrate on building consistent subsumption and disjointness axioms in these ontologies without any extra object properties.

In the case of CUB-200-2011, the attribute annotations of the dataset were used as the knowledge source to build an ontology that contains complex class axioms with additional object properties such as *hasColor*, *hasShape* etc. All four ontologies were saved in the OWL Functional Syntax format and tested for consistency via the HermiT OWL reasoner. Next, these ontologies will be used in concept embedding learning processes.

In this chapter, I introduced two main methods of learning embeddings from ontologies. The first captures the similarities between concepts and represent them as points in an embedding space. The second embeds concepts as *n*-balls using the subsumption and disjointness axioms in the ontology. I carry out experiments using the ontologies constructed in Chapter 4 and discuss the quality and the features of the resulting embeddings.

Geometrically, an *n*-ball can represent more information using both its radius and

centre values when compared to *CSim* embeddings in a vector space. For example, n-balls can clearly represent the axiom $Poodle \sqsubseteq Dog$ with the n-ball of *Dog* enclosing that of $Poodle$. But in the case of *CSim*, this subsumption is represented only by the proximity of *Dog* point to that of $Poodle$ where the position is arbitrary. Hence, I argue that n-balls are superior representations of ontology-based background knowledge in an embedding space.

In Chapter 5, I utilise these embeddings to inform a vision model and evaluate their contributions in guiding the task of few-shot image classification.

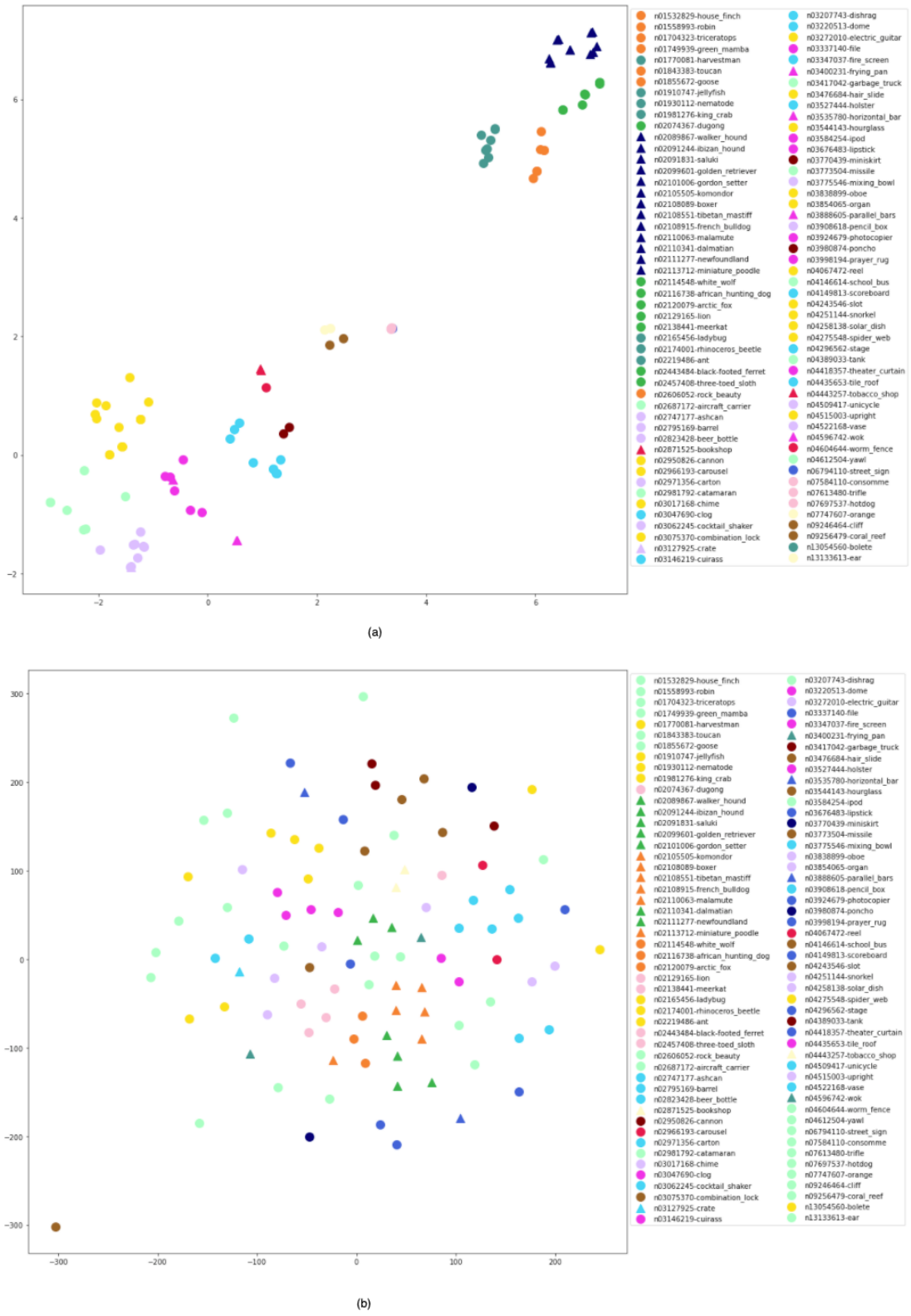


Figure 4.4: Visualisation of miniImageNet ontology embeddings. (a) *CSim* embeddings. (b) Graph-based WordNet embeddings.

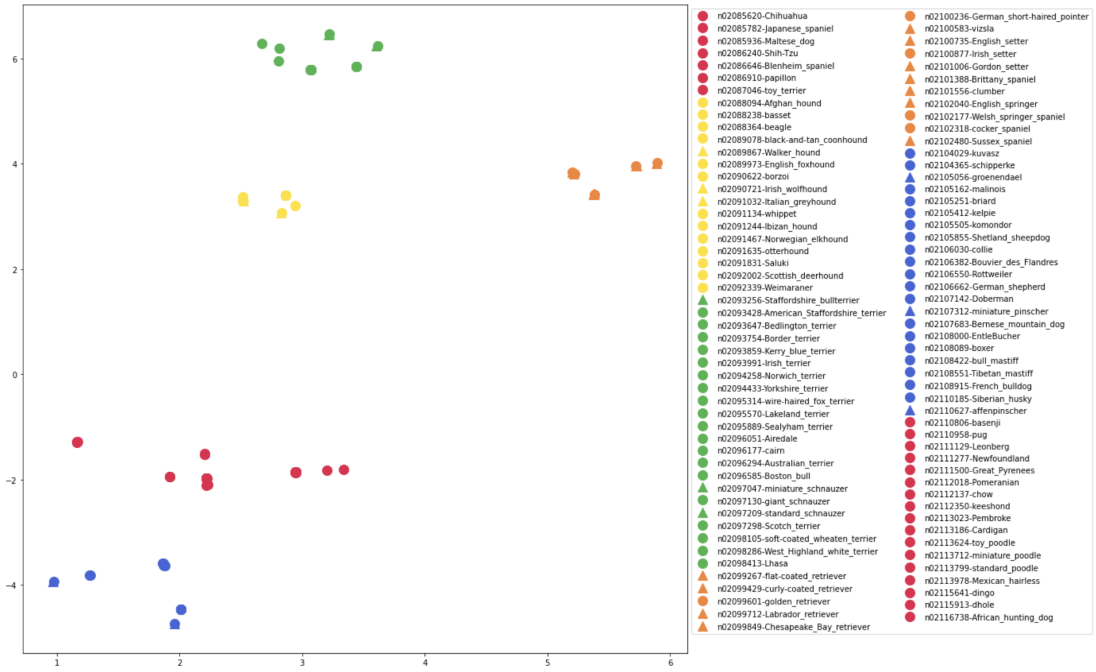


Figure 4.5: Visualisation of *CSim* embeddings on Stanford Dogs ontology.

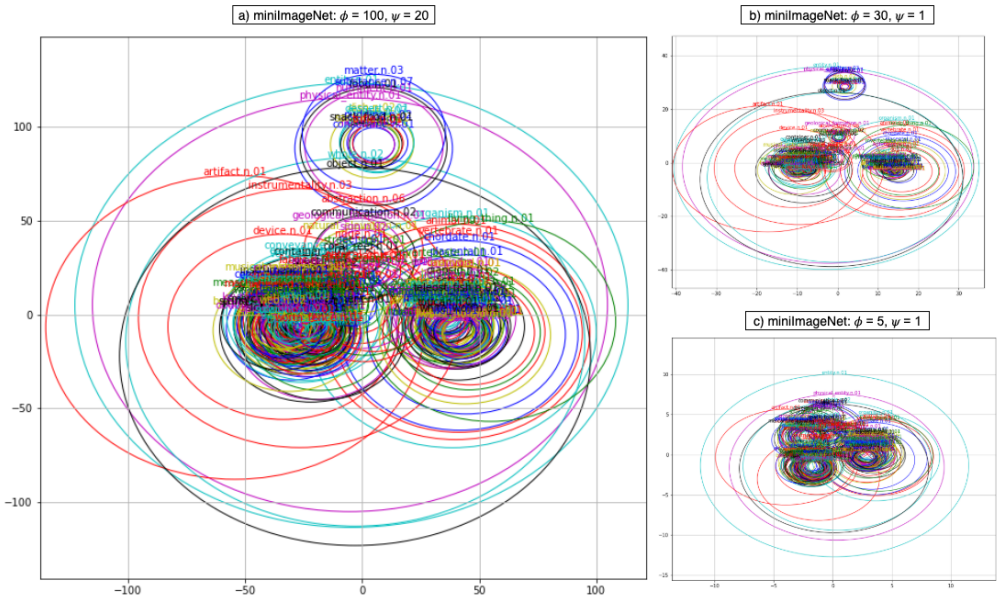


Figure 4.6: Visualisation of learnt concept embeddings from miniImageNet ontology with varying ϕ and ψ

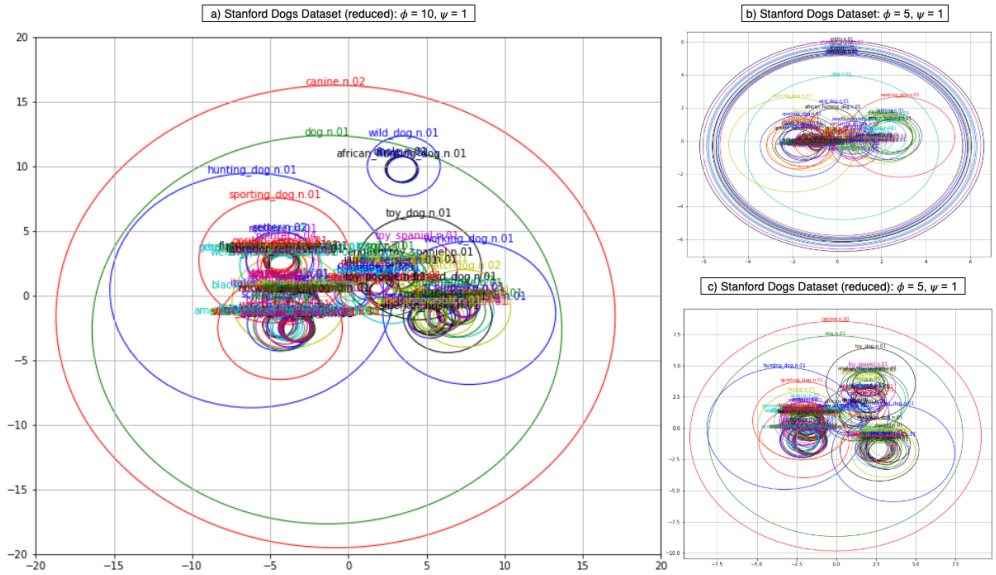


Figure 4.7: Visualisation of learnt concept embeddings from Stanford Dogs ontology with varying ϕ and ψ and reduction of classes

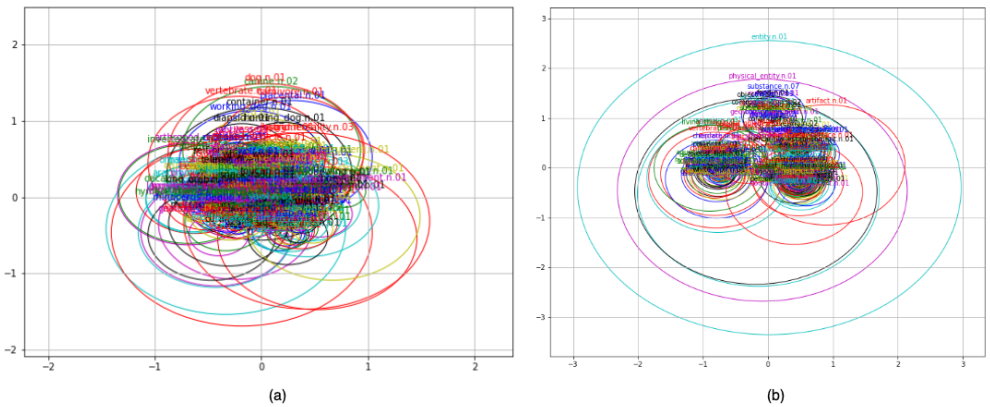


Figure 4.8: Concept embeddings learnt from miniImageNet ontology a) Using the asserted class hierarchy as input b) Using the inferred class hierarchy as input

Chapter 5

Few-shot Image Classification Informed by Concept Embeddings

In this chapter, I use the ontology-based concept embeddings proposed in Chapter 4.4 to inform a DCNN-based vision model performing the task of few-shot image classification. The goal is to evaluate the impact of integrating additional background knowledge as concept embeddings and guiding the vision model in the training and inference stages. I compare resulting classification accuracies with existing approaches in few-shot image classification. Moreover, the errors during classification are analysed for their semantic meaningfulness using the framework proposed in Chapter 5.6.

One of the research goals of this study is to investigate the contribution of background knowledge in helping a vision model train on on limited data. The existing work on few-shot image classification [CLK+19a, HGP20b] presents several benchmarks that can be used to compare performance of approaches to this end. I make use of four benchmark datasets in this study, dealing with both few-shot and fine-grained few-shot [HQDN19] image classification tasks. Fine-grained classification focuses on distinguishing classes that are visually similar (e.g., breeds of dogs [KJYFF11]), which is a harder task.

Another major consideration was the stage at which background knowledge was integrated with the vision model. I select the training stage in this case, where knowledge in the form of concept embeddings guide the loss function when mapping images to a vector space. This approach was inspired by earlier proposals of Frome et al. [FCS+13] where they used word embeddings in a similar manner. I argue that the choice to control the image mapping with background knowledge increases the transparency of the prediction space. It informs the vision model on what classes should be

mapped similarly based on the background knowledge, that in turn allows a better understanding of a prediction using the same knowledge. For example, the mapping of a *Poodle* image should be more similar to that of a *Golden_retriever* than an *Arctic_fox*. This is because both *Poodle* and *Golden_retriever* falls under *Dog* which makes them more similar to each other than to an *Arctic_fox*, that can also look visually similar to a *Dog*. In order to obtain this behaviour from a trained vision model, I argue that integrating the background knowledge during the training stage as concept embeddings is suitable approach.

The following sections introduce a framework to integrate concept embeddings with a DCNN-based vision model and evaluate its performance on the task of few-shot image classification. I present results on the semantic meaningfulness of errors and further analyse the behaviour of the overall system.

5.1 Proposed Method: ViOCE

I propose a framework named ViOCE[1] that can be used to inform a vision model with ontology-based background knowledge. It is composed of two main components: (1) a process to embed the ontology O in a vector space using the approaches proposed in Chapter 4.4, (2) a vision model to embed images as points in the same Euclidean space as the concept embeddings with a suitable arrangement, and to infer the class for a query image based on its image embedding and the concept embeddings of the candidate classes. Figure 5.1 shows the general flow of the framework with an overview of all processes and example data inputs. If a suitable ontology for the task does not exist, the ontology O for background knowledge was constructed using the class label information C_B and C_F, along with their super-class information C_H from WordNet. Next, the flow shows the two main components of ViOCE - A) Concept embedding learning process that starts with computing the inferred class hierarchy (ICH) of the input ontology that also checks the consistency of knowledge. Then it generates concept embeddings for all concepts found in the ontology via the Embedding Generator (The Embedding Generator represents any of the proposed methods used for embedding learning). B) Vision model (DCNN+MLP) training where, first the background images are used to train a base model which gets fine-tuned (only MLP) using the few-shot images to produce the final model. During both base learning and few-shot learning processes, the concept embeddings guide the learning process by setting the

[1] ViOCE is an acronym for 'Vision model informed by Ontology-based Concept Embeddings

objective of the model to project the image feature points relative to the concept embeddings representing the ground truth label of an input image.

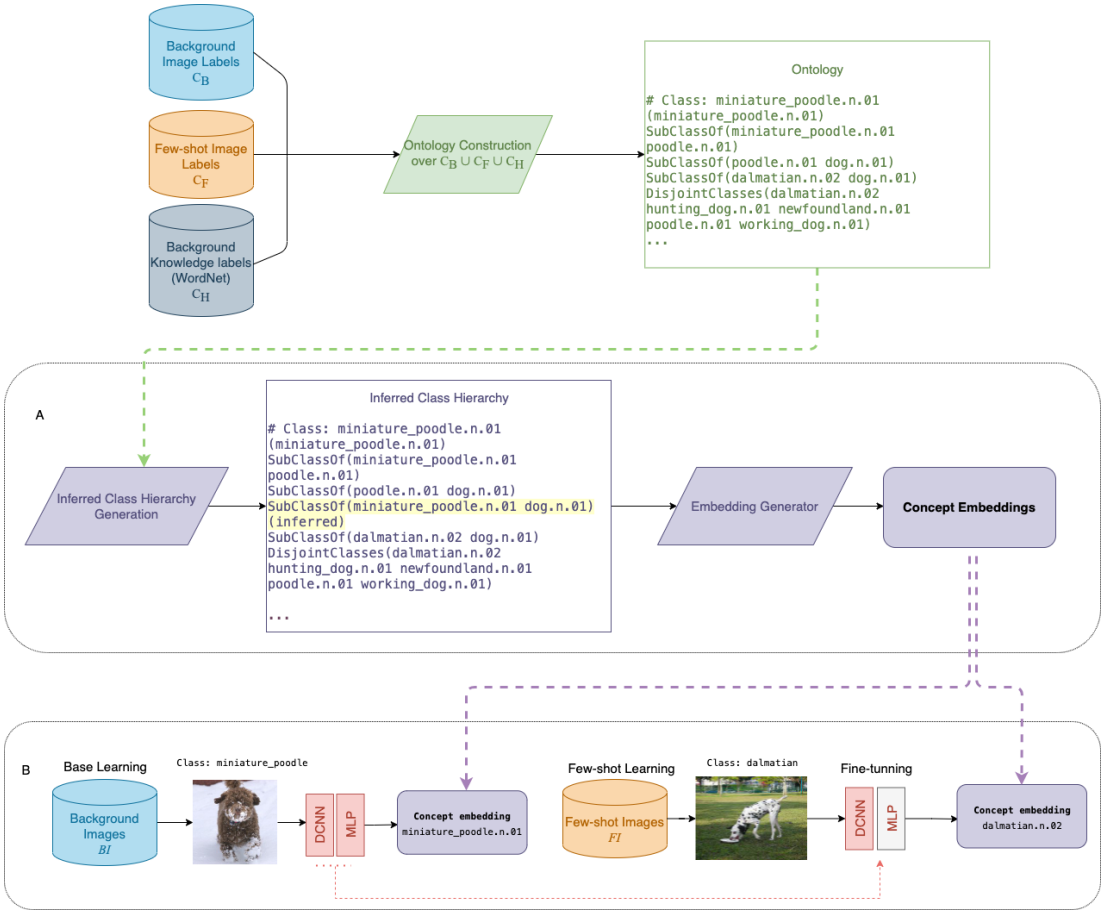


Figure 5.1: The overview of the proposed ViOCE framework

The DCNN-based vision model is trained using m background images $BI = \{(\boldsymbol{I}_i, y_i)\}_{i=1}^{m}$ (base set) from $\mathcal{K}$ classes with $y_i \in C_B = \{c_1, c_2, \ldots c_{\mathcal{K}}\}$ and s few-shot images $FI = \{(\boldsymbol{I}_i, y_i)\}_{i=1}^{s}$ (novel set) from w classes with $y_i \in C_F = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_w\}$, where $C_B \cap C_F = \emptyset$, and $\boldsymbol{I}_i$ denotes the collection of images with labels y_i from classes C, where $C = C_B \cup C_F$. In practice, I first train the DCNN and MLP from scratch by minimising the respective loss function of the vision model using the background images BI. This is called *base learning (BL)*. Then, I fine tune the MLP by using the few-shot images FI by minimising the same loss, but keep the weights of DCNN fixed. This is called *few-shot learning (FSL)*.

The few-shot success is usually assessed by how accurate a model can select a correct class from the candidate class set for a new image in C_F. This is often referred to as the w-way s-shot few-shot image classification.

When using ViOCE to integrate concept embeddings learnt in Chapter 4.4, the image embedding learning process of the vision model has to adjusted according to the form of the embedding, i.e., according to whether they are point or n-ball embeddings.

5.1.1 Image Embedding Learning with *CSim* embeddings

The vision model is composed of a base DCNN architecture coupled with a multi-layer perceptron (MLP). The DCNN computes the visual features for an image by taking its raw pixel representation as the input: $\boldsymbol{f}_i = \phi_{\mathrm{D}}(\boldsymbol{I}_i, \boldsymbol{\theta}_{\mathrm{D}})$ where $\boldsymbol{f}_i \in \mathbb{R}^d$. The MLP is responsible for mapping the visual features $\boldsymbol{f}_i$ to the n-dimensional Euclidean space where the concept embeddings sit: $\boldsymbol{h}_i = \phi_{\mathrm{M}}(\boldsymbol{f}_i, \boldsymbol{\theta}_{\mathrm{M}})$ where $\boldsymbol{h}_i \in \mathbb{R}^n$. I use $\boldsymbol{\theta}_{\mathrm{D}}$ and $\boldsymbol{\theta}_{\mathrm{M}}$ to denote the neural network parameters to be trained for the DCNN and MLP, respectively.

When using *CSim* embeddings, the idea is to identify visual features of an image (using a DCNN) so that they can be mapped (by an MLP) as an image point as close as possible to the embedding of the concept representing its ground truth class. For example, an image containing the visual features of a *Miniature_Poodle* should be mapped as close as to the *CSim* embedding of the *miniature_poodle.n.*01 concept learnt from the ontology.

To achieve this, the following pairwise ranking loss is used to optimise the network parameters:

$$l_I(\boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{\theta}_{\mathrm{M}}) = \sum_{i=1}^{m} \Big[max(0, m - \boldsymbol{e}_P \cdot \boldsymbol{h}_i + \boldsymbol{e}_Q \cdot \boldsymbol{h}_i) \Big], \tag{5.1}$$

where m is the margin constant, $\boldsymbol{e}_P$ and $\boldsymbol{e}_Q$ stand for positive and negative concept embeddings respectively, for an image i. $\boldsymbol{e}_Q$ is chosen at random out of the candidate classes such as $\boldsymbol{e}_P \neq \boldsymbol{e}_Q$.

5.1.2 Image Embedding Learning with n-ball embeddings

The vision model used with n-ball embeddings follows the same DCNN + MLP configuration as in Section 5.1.1. But a change in the objective when mapping takes place, where now the goal is to map image points to go inside the relevant n-balls representing the ground truth labels. For example, an image containing the visual features of a *Miniature_Poodle* should be mapped to be inside the radius of the n-ball representing

*miniature_poodle.n.*01 concept learnt from the ontology.

To achieve this, the pairwise raking loss is modified as below:

$$l_I(\boldsymbol{\theta}_{\mathrm{D}}, \boldsymbol{\theta}_{\mathrm{M}}) = \sum_{i=1}^{m} \Bigg[\max\left(0, \|\boldsymbol{c}_P - \boldsymbol{h}_i\|_2 - \mu r_P\right) + \sum_{Q \in C_i^{(-)}} \max(0, \nu r_Q - \|\boldsymbol{c}_Q - \boldsymbol{h}_i\|_2) \Bigg], \tag{5.2}$$

where $\mu, \nu > 0$ are hyperparameters. The set $C_i^{(-)}$ contains the negative classes defined for each image. When setting $\mu = \nu = 1$, the loss enforces $\|\boldsymbol{c}_P - \boldsymbol{h}_i\|_2 \leq r_P$, pushing the embedded image point to stay inside the n-ball of the correct class P, while $\|\boldsymbol{c}_Q - \boldsymbol{h}_i\|_2 \geq r_Q$, to stay outside the n-ball of each incorrect class Q. The hyperparameters μ and ν are placed to control the intensity of this effect, e.g., $\mu < 1$ requiring to lie closer to the center which makes the task harder.

With n-ball embeddings, a specification crucial to learning performance is the selection of concepts in $C_i^{(-)}$. Following the notion of "hard negatives" in [MJWG12], we select "hard negatives" based on proximity. For example, the "poodle" concept is more similar to "golden retriever" in contrast to the "street sign", therefore it is more challenging to distinguish between "poodle" and "golden retriever". Hence, we identify "golden retriever" as the hard negative of "poodle". Specifically, we evaluate similarities between concepts by Euclidean distances between the centre vectors of their corresponding n-balls, and perform k-means clustering based on these. After clustering the centre vectors of the leaf concepts (image classes), for each image class, all the other image classes from the same cluster are treated as the "hard negatives" and are included in $C_i^{(-)}$.

5.1.3 Model Inference

Following the vision model training process, concept embeddings are used when making predictions with the test set of FI (FI_{te}) as well. With both *CSim* and n-ball embeddings, given a new query image $\boldsymbol{I}$, I compute its image embedding by $\boldsymbol{h}(\boldsymbol{I}) = \phi_{\mathrm{M}}(\phi_{\mathrm{D}}(\boldsymbol{I}, \boldsymbol{\theta}_{\mathrm{D}}), \boldsymbol{\theta}_{\mathrm{M}})$ using the respective trained vision model.

During inference with *CSim* embeddings, I select the closest concept embedding $\boldsymbol{e}_i$ to $\boldsymbol{h}(\boldsymbol{I})$ out of the candidate classes by $\arg\min_{i \in w} \|\boldsymbol{e}_i - \boldsymbol{h}(\boldsymbol{I})\|$ as the prediction. This is following the objective that the vision model should map image points as close as to

the respective concept embeddings of the ground truth classes.

In the case of n-ball embeddings, I choose the closest candidate class n-ball centre $\boldsymbol{c}_i$ to $\boldsymbol{h}(\boldsymbol{I})$ by $\arg\min_{i \in w} \|\boldsymbol{c}_i - \boldsymbol{h}(\boldsymbol{I})\|$, as the prediction. According to this, $\boldsymbol{h}(\boldsymbol{I})$ being inside the n-ball of the predicted class is not always a necessity.

5.2 Experiments

During all the experiments, ResNet50 [HZRS16c] architecture was chosen to be the base network and the MLP was composed of 5 layers with sizes of 2048, 1024, 512, 512, and 300 (or 100 in one instance of *CSim* embeddings), respectively. The last layer size is determined by the dimensionality of the concept embeddings used. Except the last two stages, each MLP layer performs batch normalisation followed by a linear layer with ReLU activation. We use Tanh as activation in the fourth stage and the output from the linear layer at the fifth stage was taken without activation to match with the distribution of the concept embeddings. Stochastic gradient descent was used with a momentum value of 0.9 and a learning rate of 0.001, where a learning rate decay was performed every 10 epochs with a factor of 0.1. Base learning (*BL*) was carried out for only 30 epochs with every dataset along with a batch size of 64.

We perform w-way s-shot image classification in this study, where w is the number of candidate classes in a given few-shot task and s is the number of images available per class during training. The w unseen classes and s images per class are randomly selected for each task. Analogous to existing studies [TWK$^+$20], we conduct experiments for $w = \{5, 20\}$ and $s = \{1, 5\}$.

At the stage of few-shot learning (FSL), the same gradient decent parameters were used from *BL* but without decay in learning rate. FSL was carried out for 100 epochs with a batch size of 16. All experiments were iterated 10 times and the average accuracies were reported.

In terms of dataset splits, following the configuration in [HQDN19] with miniImageNet dataset, 80 and 20 classes were allocated for *BI* and FI, respectively. This is analogous to the 64 class meta-training, 16 class meta-validation and 20 class meta-testing split in some few-shot learning studies [RL16a, TWK$^+$20], if the training and validation classes were combined. According to the specification of tieredImageNet, for *BI* I use a set of 448 classes belonging to 26 higher-level categories. For FI, a set of 160 classes were selected belonging to 8 higher-level categories. Following the configuration in [HQDN19] with the Stanford Dogs dataset, I randomly select a subset

of 100 classes and randomly subdivide them into 80 *BI* classes and 20 *FI* classes. The 200 classes of CUB-200-2011 were randomly divided into 150 classes of *BI* and 50 classes and *FI*. were randomly divided into 150 classes of *BI* and 50 classes and *FI*.

5.2.1 Experimental Setup for *CSim* Embeddings

In Equation 5.1, m was set to 30 during *BL* and 100 during *FSL*. A higher m makes the task of minimising of the objective function harder, helping the vision model learn the mapping with a few examples during FSL.

5.2.2 Experimental Setup for n-ball Embeddings

In Equation 5.2, μ and ν parameters were set to 0.1 and 1 respectively. k values for the selection of hard negatives were set to 15 for miniImageNet and Stanford Dogs and 26 for tieredImageNet.

During *FSL*, μ was kept at 0.1, the same as *BL*, but ν was increased to 10 to allow the negative samples to have more effect. Especially with 1-shot learning, this was found to be helpful to make the training more effective. k was set to 3 during 5-way classifications and 15 during 20-way.

This setup was used when training with both n-ball and multi-relational n-ball embeddings.

5.3 Results and Comparative Analysis

5.3.1 Few-shot Classification: *CSim* and n-ball Embeddings

Table 5.1 shows the 5-way 1-shot and 5-shot performance comparisons with state-of-the-art few-shot image classification approaches on miniImageNet and tieredImageNet datasets. Additionally, I include the performance of several standard vision models in the same 5-way setting that are not designed for few-shot learning, namely, ResNet [HZRS16c], SqueezeNet [IHM$^+$16], VGG [SZ14] and DenseNet [HLVDMW17]. Also as a control experiment, I add results of a vision model trained in ViOCE with randomly generated point embeddings without any background knowledge.

It can be seen that ViOCE generally surpasses the performance of all other approaches in every 5-way task on both the datasets while achieving $>$90% accuracy

Table 5.1: 5-way 1-shot and 5-shot accuracy comparison with existing approaches using miniImageNet and tieredImageNet benchmarks. Accuracies are reported with 95% confidence intervals.

Model	miniImageNet 5-way		tieredImageNet 5-way	
	1-shot (%)	**5-shot (%)**	**1-shot (%)**	**5-shot (%)**
ResNet [HZRS16c]	20.89	24.90	-	-
SqueezeNet [IHM$^+$16]	19.86	21.39	-	-
VGG [SZ14]	21.93	67.70	-	-
DenseNet [HLVDMW17]	27.04	32.49	-	-
MAML [FAL17]	48.70 ± 1.84	63.11 ± 0.92	51.67 ± 1.81	70.30 ± 1.75
Matching Networks [VBL$^+$16a]	43.56 ± 0.84	55.31 ± 0.73	-	-
IMP [ASST19]	49.20 ± 0.70	64.7 ± 0.70	-	-
Prototypical Networks [SSZ17]	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
TAML [AJQS18]	51.77 ± 1.86	66.05 ± 0.85	-	-
SAML [HHC$^+$19]	52.22 ± n/a	66.49 ± n/a	-	-
GCR [LLX$^+$19]	53.21 ± 0.80	72.34 ± 0.64	-	-
KTN (Visual) [PLZ$^+$19]	54.61 ± 0.80	71.21 ± 0.66	-	-
PARN [WLGJ19]	55.22 ± 0.84	71.55 ± 0.66	-	-
Dynamic Few-shot [GK18b]	56.20 ± 0.86	73.00 ± 0.64	-	-
Relational Networks [SYZ$^+$18b]	50.44 ± 0.82	65.32 ± 0.70	54.48 ± 0.93	71.32 ± 0.78
R2D2 [BHTV18b]	51.2 ± 0.6	68.8 ± 0.1	-	-
SNAIL [MRCA17]	55.71 ± 0.99	68.88 ± 0.92	-	-
AdaResNet [MYMT18]	56.88 ± 0.62	71.94 ± 0.57	-	-
TADAM [ORL18]	58.50 ± 0.30	76.70 ± 0.30	-	-
Shot-Free [RBS19]	59.04 ± n/a	77.64 ± n/a	63.52 ± n/a	82.59 ± n/a
TEWAM [QSL$^+$19]	60.07 ± n/a	75.90 ± n/a	-	-
MTL [SLCS19]	61.20 ± 1.80	75.50 ± 0.80	-	-
Variational FSL [ZZN$^+$19]	61.23 ± 0.26	77.69 ± 0.17	-	-
MetaOptNet [LMRS19]	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
Diversity w/ Cooperation [DSM19]	59.48 ± 0.65	75.62 ± 0.48	-	-
Fine-tuning [DCRS19b]	57.73 ± 0.62	78.17 ± 0.49	66.58 ± 0.70	85.55 ± 0.48
LEO-trainval [RRS$^+$18b]	61.76 ± 0.08	77.59 ± 0.12	66.33 ± 0.05	81.44 ± 0.09
Embedding-distill [TWK$^+$20]	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
PAL [MXH$^+$21]	69.37 ± 0.64	84.40 ± 0.44	72.25 ± 0.72	86.95 ± 0.47
ViOCE (Random point embeddings)	40.47 ± 0.90	77.93 ± 0.69	-	-
ViOCE (WordNet embeddings [SBRS18])	67.30 ± 0.85	91.03 ± 0.51	-	-
ViOCE (Owl2vec embeddings [HMCJR19b])	52.65 ± 0.62	73.25 ± 0.32	-	-
ViOCE (EL embeddings [KLWYH19])	61.12 ± 0.72	82.41 ± 0.12	-	-
ViOCE (*CSim* embeddings)	**68.41 ± 0.84**	90.81 ± 0.52	-	-
ViOCE (n-ball embeddings)	65.71 ± 0.13	**93.65 ± 0.07**	**73.4 ± 0.13**	**88.95 ± 0.09**

in miniImageNet 5-shot task. The model informed by random point embeddings performs worse compared to all other models informed by concept embeddings. One noticeable result is in the 5-way 1-shot setting, ViOCE with both WordNet embeddings and *CSim* embeddings surpass the performance of ViOCE with n-ball embeddings. This implies that learning with a very small number of examples (in this case just 1 image per class) is a harder task with n-ball embeddings than with point form embeddings. To further understand this effect I visualise the behaviour of image embeddings

and relevant *n*-balls of the candidate classes during an instance of 5-way 1-shot and 5-shot few-shot classification training. Figure 5.2 demonstrates this, where the projected image points are shown in blue and the candidate concept embeddings are shown in red. The objective of the vision model here was to embed an image as a point inside its label's *n*-ball. Comparing Figure 5.2 (a) and (b), it can be seen that this is harder to achieve in the 1-shot case than the 5-shot. Even though the distribution of points is somewhat directed towards the *n*-balls during 1-shot, they do not reach the inside as much as with 5-shot. The higher accuracies of 5-shot classification in Table 5.1 reflect this behaviour. These observations were consistent with all the other datasets when using *n*-ball embeddings.

Next, the study further extends the evaluation with the miniImageNet dataset to the task of 20-way 1-shot and 5-shot classification. Having all the 20 few-shot classes should offer a bigger challenge to the model, having to distinguish between more possible classes with a few examples. Table 5.2 presents the result comparison for this task with other state-of-the-art few-shot classification models. I add a vision model trained with randomly generated point embeddings in this task as well.

Table 5.2: 20-way 1-shot and 5-shot accuracy comparison with existing approaches using miniImageNet dataset.

Model	**miniImageNet 20-way**	
	1-shot (%)	**5-shot (%)**
MAML [FAL17]	16.49	19.29
Meta LSTM [VBL$^+$16a]	16.70	22.69
Matching Networks (Vinyals et al.)	17.31	26.06
Meta SGD [LZCL17]	17.56	28.92
Deep Comparison Network [ZQS$^+$18]	32.07	47.31
TIM-GD [BMR$^+$20]	39.30	59.50
ViOCE (Random point embeddings)	25.51	46.08
ViOCE (WordNet embeddings [SBRS18])	34.81	58.83
ViOCE (*CSim* embeddings)	31.63	54.75
ViOCE (*n*-ball embeddings)	**48.02**	**84.13**

It can be seen that ViOCE with *n*-ball embeddings surpasses the performance of the existing approaches in both 20-way 1-shot and 5-shot tasks. Referring to Figure 5.2 (c) and (d) visualising the behaviour of image points and candidate *n*-ball concept embeddings during 20-way *FSL*, it can be seen that the task of mapping images towards the

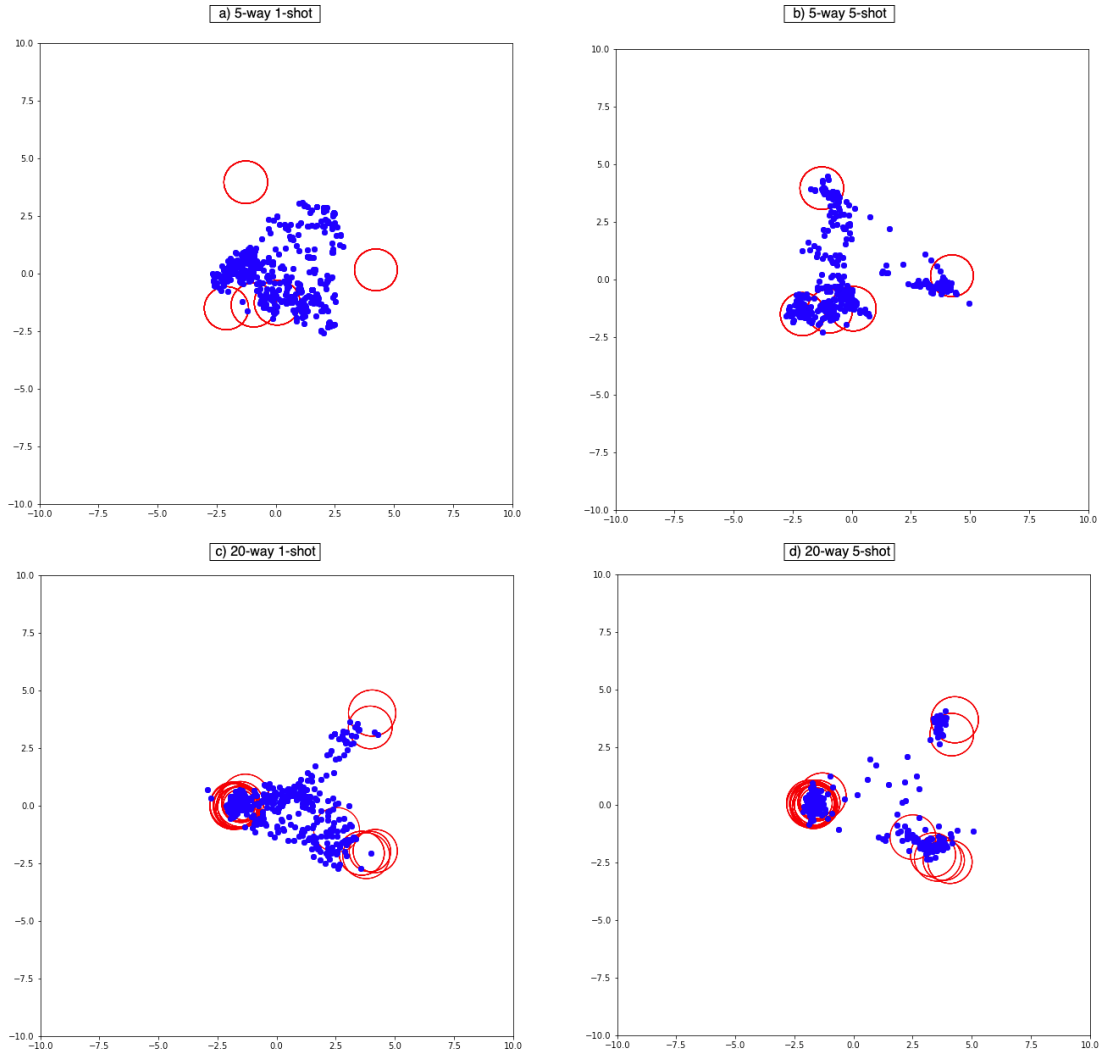


Figure 5.2: Visualisation of projected image feature points (in blue) in the vicinity of target concept embeddings (in red) during instances of 5-way and 20-way few-shot training of ViOCE using miniImageNet.

n-balls has become somewhat easier for the vision model in this case, especially during 5-shot. I see this as a result of having more negative samples to direct the feature points during training in the case of 20 candidate classes.

Moreover with Stanford Dogs dataset, I produce fine-grained few-shot image classification results for ViOCE and compare them with several existing few-shot approaches using the same dataset. Table 5.3 shows the results for both 5-way and 20-way settings with Stanford Dogs.

Table 5.3: Fine-grained few-shot image classification accuracies of ViOCE compared with exiting approaches.

Model	Stanford Dogs 5-way		Stanford Dogs 20-way	
	1-shot (%)	**5-shot (%)**	**1-shot (%)**	**5-shot (%)**
MAML [FAL17]	31.52	59.66	-	-
Meta-Learner LSTM [RL16b]	38.37	53.65	-	-
Matching Nets FCE++ [VBL$^+$16a]	46.01	57.38	-	-
DN4-DA (k=1) [LWX$^+$19]	45.73	66.33	-	-
MATANet [CLLC20]	55.63	70.29	-	-
MML [CLLC21]	59.05	75.59	-	-
Standard ResNet50 [HZRS16c]	36.09	71.13	5.4	7.1
ViOCE (WordNet embeddings [SBRS18])	**88.70**	93.32	52.57	71.67
ViOCE (*CSim* embeddings)	87.45	90.68	44.69	65.55
ViOCE (n-ball embeddings)	74.76	**95.45**	**60.51**	**86.29**

Overall with fine-grained few-shot image classification, ViOCE produces superior accuracies in the tasks with all embedding types when compared to existing approaches. The observation from Table 5.1 is consistent in this case as well, where the point embeddings perform significant better in the 5-way 1-shot setting. This is again implying the difficulty of learning with n-balls with just 1 image per class. But this effect is overcome in the 20-way 1-shot setting as the increase in the number of candidate class seems to have helped the FSL process with n-balls.

In general, the higher differences in the performance of all ViOCE approaches in this task when compared to the existing methods imply the supportive role of background knowledge during the challenging task of identifying similar-looking classes with a few examples.

5.3.2 Few-shot Classification: Multi-relational n-ball Embeddings

Multi-relational n-ball embeddings were evaluated separately on the CUB-200-2011 dataset and compared with other existing state-of-the-art approaches that perform few-sot image classification on the same dataset. Table 5.4 shows the accuracies of ViOCE using the three types of embeddings proposed in Section 4.8.

It can be seen that ViOCE with all three embedding methods produces accuracies greater than 80% in both 5-way 1-shot and 5-shot tasks. ViOCE (FO-EM) shows the best performance, producing accuracies above 90%, while surpassing the state-of-the-art in the 5-way 5-shot task.

Table 5.4: 5-way 1-shot and 5-shot image classification on CUB-200-2011

Methods	5-way classification	
	1-shot (%)	5-shot (%)
Baseline++ [CLK+19b]	69.55 ± 0.89	85.17 ± 0.50
MAML [FAL17]	70.32 ± 0.99	80.93 ± 0.71
ProtoNet [SSZ17]	72.99 ± 0.88	86.64 ± 0.51
Matching Networks [VBL+16b]	73.49 ± 0.89	84.45 ± 0.58
S2M2_R [MKS+20]	80.68 ± 0.81	90.85 ± 0.44
Transfer+SGC [HGP20a]	88.35 ± 0.19	92.14 ± 0.10
PT+MAP [HGP20c]	**91.55 ± 0.19**	93.99 ± 0.10
ViOCE (FO-EM)	90.34 ± 0.18	**96.48 ± 0.11**
ViOCE (TF-EM)	81.64 ± 0.23	88.23 ± 0.19
ViOCE (SP-EM)	85.37 ± 0.21	94.14 ± 0.16

5.4 ViOCE vs Knowledge Encoded as Class Labels

An alternative simpler approach to inform a vision architecture with background knowledge is setting up the vision task as multi-label classification, where the labels of the images are extended using the background knowledge. It can be argued that the extra information in the label space can then be transferred into the vision model during training [WYM+16]. I explore how this technique compares with ViOCE by setting up an experiment to perform the multi-label classification with ViOCE using *n*-ball embeddings and a standard ResNet [HZRS16c] architecture. I only use ViOCE with *n*-ball embeddings here, as modifying the *CSim* embedding setup to perform multi-label classification is challenging.

In multi-label classification, for a prediction to be correct, a model should output all of the relevant labels given an input image. To this end, I modify the inference procedure of ViOCE with *n*-ball embeddings, where a prediction is taken only when an image point is inside a *n*-ball. During inference, the candidate classes are extended with all the *n*-balls of their more general classes, i.e., the prediction can be several concepts. For example, an image point mapped inside the *Poodle* *n*-ball can predict both *Poodle* and *Dog*, since the *Poodle* *n*-ball is enclosed by the *Dog* *n*-ball.

This experiment used the 80 training classes of miniImageNet dataset with 500 images per class and each label was extended with all its super-classes according to the miniImageNet ontology. The ResNet model was trained using stochastic gradient descent with a learning rate of 0.1 and a binary cross-entropy loss coupled with a

sigmoid layer to be assessed against a one-hot encoded vector in the label space. The model was trained for 100 epochs with a batch size of 8.

The results recorded a training accuracy of 47.33% and a testing accuracy of 75.82% for the standard ResNet vision model. Whereas, ViOCE reported 79.34% and 95.01% as training and testing accuracies respectively. Even though the standard model attains considerable accuracy and an effective generalisability, ViOCE achieves much better performance in the multi-label classification task. We argue that this is caused by the effectiveness and transparency of the knowledge infusion technique using concept embeddings in ViOCE, whereas providing more information in the form of labels can be inefficient and noisy.

5.5 Ablations Studies

5.5.1 Random Hard Negatives with n-ball Embeddings

A vital part of the technique proposed to train vision model with n-ball embeddings is the selection of hard negatives as discussed in Section 5.1.2. I look at the importance of choosing hard negatives via the proposed k-means clustering technique by comparing it with a random selection of hard negatives. Taking the 20-way 5-shot setting with the miniImageNet dataset, I setup an experiment to choose 5, 10 and 15 classes at random from the 20 classes as hard negatives during the training process. I select the 20-way case because it gives us enough classes with similarities and dissimilarities to understand how the selection of hard negatives can be important. All other experimental details remains the same as in Section 5.2.2. The results reported accuracies of 67.32%, 75.37% and 79.14% for the cases of 5, 10 and 15 random hard negatives respectively. This compares with the accuracy of 84.13% in Table 5.2 for ViOCE with the proposed method of hard negative selection. This empirically shows that the approach of selecting the most similar classes as negatives via clustering can help guide the mapping of the vision model.

5.6 Semantically Meaningful Errors

This chapter introduces the notion of semantically meaningful errors in an image classification task and proposes a framework which is used to capture them in the proceeding few-shot image classification experiments. Existing work in image classification

[HZRS16c, KSH12b] expects the prediction from the vision model to be the ground truth label of the image and otherwise it is considered incorrect. But I argue that not all classification errors are equal, especially when evaluating a model for its ability to gain a conceptual understating about objects. Prediction of a comparatively similar concept to the ground truth concept should be "less wrong" than predicting a largely dissimilar concept. For example, classifying a *Poodle* image as a *German_Shepherd* should be less wrong than classifying it as a *Fish*. So, according to the similarity between the concepts in the prediction space there should be a way to capture the degree of error during evaluation.

Better understanding of errors can be useful when evaluating how a trained model is behaving during a classification task. Information such as frequently misclassified class pairs provides insights into biases captured during the training phase. Although misclassifications between very similar fine-grained classes can sometimes be expected, confusions between totally dissimilar classes should be considered as bigger errors. Identification of these guides the revisiting of the training data and the background knowledge used during training.

The use of background knowledge about the classes in the classification task is the factor that enables the measure of errors in this case. To this end, I have proposed a systematic framework to measure the degree of error in few-shot image classification results using the class hierarchy of the ontologies used as background knowledge.

5.7 Existing Evaluation Method

The main metric during the existing evaluation of few-shot image classification is found to be the score of accuracy when classifying a test set of a task [CLK$^+$19a]. Here, the accuracy is governed by the portion of the correct predictions out of all the predictions of a model. A correct prediction is an exact match between the predicted class and the ground truth class, otherwise it is considered an error.

This framework does not contain a mechanism to understand more about the errors, which is important because some errors can be better than the others. But existing approaches that does not incorporate background knowledge about the classes also lacks the extra knowledge needed to evaluate these errors meaningfully. Hence, I propose an extension to the evaluation framework in this study, that also measures the degree of errors during a few-shot image classification task.

5.8 Proposed Framework

Along with the existing accuracy calculation, the proposed framework is designed to determine the semantic meaningfulness of the predicted class with respect to the ground truth label class in case of an error in prediction among the candidate classes. The similarity value between these two classes is taken as the degree of error in classification, where a higher similarity means a lower degree of error, i.e., a more semantically meaningful error.

Figure 5.3 shows the general flow of the similarity computation process of the proposed framework that outputs the degree of error. Given the prediction of the vision model for an input image, first it is compared with the ground truth class to determine if it is an error. If so, both the predicted and the ground truth classes are fed into the similarity computation process that takes in an ontology-based similarity measure as another input. This measure determines the similarity between the two classes, the score of which becomes the output degree of error during that classification instance.

The framework is open to any choice of a similarity measure according to the information available from the background knowledge used in a task. In this study, I propose the feature-based similarity between concepts of an ontology (introduced in Section 2.2.2) as the default measure to be used in this error analysis. The type of feature-based similarity, whether it should be atomic or subconcept similarity, is decided according to the knowledge available in the ontology in a given case. For example, considering the Birds Ontology (Section 4.3.2), since it provides more features of the bird species in the class expressions rather than via the class hierarchy, subconcept similarity (Section 2.2.2.2) would be a better choice than the atomic similarity (Section 2.2.2.1). Figures 5.4 (a) and 5.4 (b) are two snapshots of the Birds Ontology showing the information available in the class hierarchy and the class expressions, respectively.

For each trained vision model during the experiments in Chapter 5, I determine the semantic meaningfulness of its classification errors using this framework. For errors when classifying a test set, I calculate the $mean, max$ and $mode$ values of the feature-based similarities between the predicted and the ground truth classes. Provided that more similar classes have a higher similarity score between them according to the feature-based similarity measure, the *mean* value determines how wrong the predictions are on average during the given classification task. The *max* determines the most similar pair of classes that has resulted in an error, while the $mode$ determines the hardest pair to distinguish for the model.

Table 5.5 shows an example format with demo results on how the error analysis

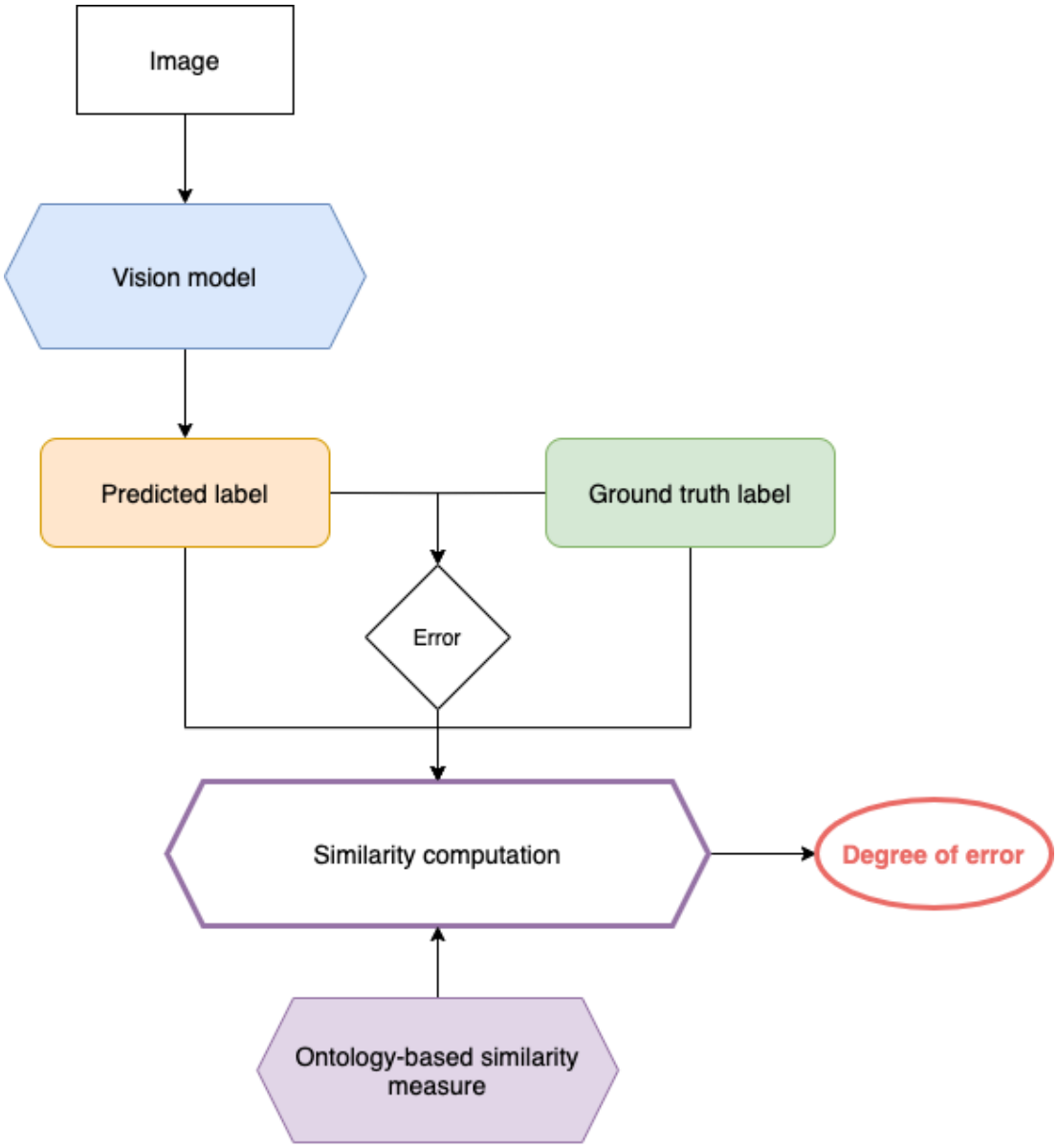
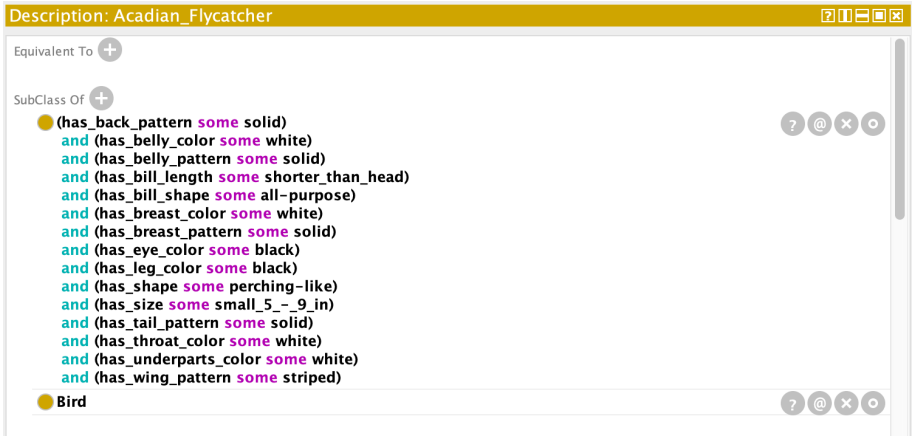


Figure 5.3: The overall flow of the proposed framework to compute the degree of error during a classification error according to a ontology-based similarity measure.

will be help to understand the difference in the behaviour of two vision models. It can be seen that even though Model 1 has higher accuracy in the classification task than Model 2, the *mean* of similarity scores during the errors of Model 1 is lower than that of Model 2. This means that Model 1 tends to misclassify between classes that more dissimilar compared to the errors of Model 2. Hence, Model 2 makes better errors than Model 1 even with less accuracy overall. Also, looking at the *max* and *mode* values it can be further seen that the highest similarity and the most frequent similarity during

(a) Class hierarchy of the Birds Ontology



(b) The set of axioms defining the bird class *Acadian_Flycatcher*

Figure 5.4: A snapshot of the Birds Ontology

the errors in Model 2 is higher than that of Model 1, again implying the better errors of Model 2.

Table 5.5: An example of semantically meaningful error analysis results during a classification task.

Classification task		
Method	**Overall Accuracy (%)**	**Semantically Meaningful Error Analysis (SMEA)**
Model 1	85	Mean - 0.8 Max - 2.3 Mode - 2.1
Model 2	83	Mean - 1.2 Max - 2.6 Mode - 2.3

5.9 Semantically Meaningful Error Analysis (SMEA)

The Semantically Meaningful Error Analysis (SMEA) was carried on the few-shot image classification results using the framework proposed in Chapter 5.6. SMEA captures the degree of error in when classifying the test set FI_{te} in each case of ViOCE, according to the class hierarchy information of the labels. This gives the opportunity to understand the properties of the trained vision model in terms of the average degree of error, most frequent errors and the hardest combinations of classes to classify.

5.9.1 SMEA: *CSim* and n-ball Embeddings

For ViOCE models informed with *CSim* and n-ball embeddings, I use the atomic similarity between concepts according to the taxonomy information (as discussed in Section 2.2.2.1) to capture the semantic meaningfulness of errors when performing few-shot image classification on miniImageNet, tieredImageNet and Stanford Dogs datasets.

Table 5.6 shows the results of SMEA during 5-way few-shot image classification of the different vision models. Comparing with the classification accuracy results of the same models, it can be seen that the mean values increase as the accuracies increase. This means that the errors of the better performing models are predicting more similar classes to the ground truth label (representative by the higher atomic similarity between predicted and label classes). The max values are consistent among all datasets,

Table 5.6: Results of SMEA using atomic similarity during 5-way classification on-miniImageNet, tieredImageNet and Stanford Dogs datasets.

Methods	**miniImageNet 5-way**			
	1-shot		**5-shot**	
	Accuracy (%)	**SMEA**	**Accuracy (%)**	**SMEA**
ViOCE (WordNet embeddings)	67.30	Mean - 0.34 Max - 0.89 Mode - 0.71	91.03	Mean - 0.58 Max - 0.89 Mode - 0.71
ViOCE (CSim embeddings)	68.41	Mean - 0.37 Max - 0.89 Mode - 0.71	90.81	Mean - 0.54 Max - 0.89 Mode - 0.68
ViOCE (n-ball embeddings)	65.71	Mean - 0.31 Max - 0.89 Mode - 0.68	93.65	Mean - 0.67 Max - 0.89 Mode - 0.89
	tieredImageNet 5-way			
ViOCE (n-ball embeddings)	73.40	Mean - 0.48 Max - 0.90 Mode - 0.61	88.95	Mean - 0.62 Max - 0.90 Mode - 0.61
	Stanford Dogs 5-way			
ViOCE (WordNet embeddings)	88.70	Mean - 0.76 Max - 0.90 Mode - 0.72	93.32	Mean - 0.80 Max - 0.90 Mode - 0.72
ViOCE (CSim embeddings)	87.45	Mean - 0.74 Max - 0.90 Mode - 0.72	90.68	Mean - 0.78 Max - 0.90 Mode - 0.72
ViOCE (n-ball embeddings)	74.76	Mean - 0.69 Max - 0.90 Mode - 0.65	95.45	Mean - 0.83 Max - 0.90 Mode - 0.80

meaning the models always made an error classifying between two classes that are very similar. The mode values also imply the same outcome where mostly similar classes (higher atomic similarity) are harder to classify for all the models.

Figures 5.5, 5.6 and 5.7 show examples of misclassified pairs according to the SMEA analysis for all three datasets.

5.9.2 SMEA: Multi-relational n-ball Embeddings

With Multi-relational n-ball Embeddings, I face a challenge when using the atomic similarity measure with the Birds Ontology. This is due to the reason that it does not provide information on a rich hierarchical structure for the bird species (all bird classes are defined under one superclass 'Bird'). I use the subconcept similarity (discussed in Section 2.2.2.2) when analysing the semantic meaningfulness of errors during the few-shot classification on CUB-200-2011. Table 5.7 shows the SMEA results during 5-way

(a) (b)

Figure 5.5: Example pair of misclassified classes from the miniImageNet dataset with atomic similarity of 0.89 (a) *Ibizan_hound* (b) *Saluki*

(a) (b)

Figure 5.6: Example pair of misclassified classes from the tieredImageNet dataset with atomic similarity of 0.90 (a) *Tiger_shark* (b) *Great_white_shark*

(a) (b)

Figure 5.7: Example pair of misclassified classes from the Stanford Dogs dataset with atomic similarity of 0.90 (a) *Curly_coated_retriever* (b) *Gordon_setter*

1-shot and 5-shot classification.

Table 5.7: Results of SMEA using subconcept similarity during 5-way classification CUB-200-2011

Methods	**CUB-200-2011 5-way**			
	1-shot		**5-shot**	
	Accuracy (%)	**SMEA**	**Accuracy (%)**	**SMEA**
ViOCE (FO-EM)	90.34	Mean - 0.37 Max - 0.66 Mode - 0.61	96.48	Mean - 0.55 Max - 0.66 Mode - 0.61
ViOCE (TF-EM)	81.64	Mean - 0.33 Max - 0.66 Mode - 0.61	88.23	Mean - 0.36 Max - 0.66 Mode - 0.61
ViOCE (SP-EM)	85.37	Mean - 0.35 Max - 0.61 Mode - 0.61	94.14	Mean - 0.42 Max - 0.66 Mode - 0.61

According Table 5.7 it can be observed that the best performing method, ViOCE (FO-EM), has the highest mean in the subconcept similarity compared to the other models. This indicates that the similarity between the errors and the ground truth is high on average. Overall, the max indicates that the worst errors made by all models are between pairs that are very similar (e.g., similarity = 0.66). The mode which indicates the most frequent error is consistent among all models where subconcept similarity = 0.61. An example for a pair of bird classes with these values is *Lazuli_Bunting* and *Painted_Bunting*. Figure 5.8 shows images of these two classes. They are the hardest to distinguish for each model.

Furthermore, I perform 50-way 5-shot classification using all 50 classes of FI to check the consistency of the above observations on errors during a classification task with higher number of classes. This is not a standard benchmark task on CUB-200-2011, hence I do not compare the accuracies with existing approaches. The results for both few-shot image classification accuracy and SMEA are shown in Table 5.8. Again, it can be seen that the mean of subconcept similarity increases as the accuracies increase, indicating that higher similarities among classes make the classification harder. The max values indicate that all models struggle with the most similar pair of classes out of the 50 and the mode values point towards classes that are the hardest to classify. Two example pairs of birds according to these values were found to be $\{Brandt\_Cormorant, Pelagic\_Cormorant\}$ and $\{Forsters\_Tern, Common\_Tern\}$. Some images of these classes are shown in Figure 5.9.

(a) (b)

Figure 5.8: Two bird classes that are the hardest to distinguish during 5-way 1-shot and 5-shot classification. (a) *Lazuli_Bunting* (b) *Painted_Bunting*

5.10 Discussion

The proposed framework in this Chapter evaluates the errors during the inference stage of a vision model. I argue that it is important to understand the behaviour of the errors separate from the correct predictions in an informed machine learning approach. This way a vision model is evaluated not only according to its prediction accuracy, but also according to the semantic meaningfulness of its errors. A good model is expected to achieve high accuracy together with high meaningfulness in errors, especially with the integration of background knowledge.

Table 5.8: Results on 50-way 5-shot accuracy and SMEA during 50-way classificationon CUB-200-2011

Methods	**50-way 5-shot (%)**	**SMEA**
ViOCE (FO-EM)	75.83	Mean - 0.88 Max - 0.96 Mode - 0.96
ViOCE (TF-EM)	70.24	Mean - 0.81 Max - 0.96 Mode - 0.85
ViOCE (SP-EM)	73.19	Mean - 0.86 Max - 0.96 Mode - 0.96

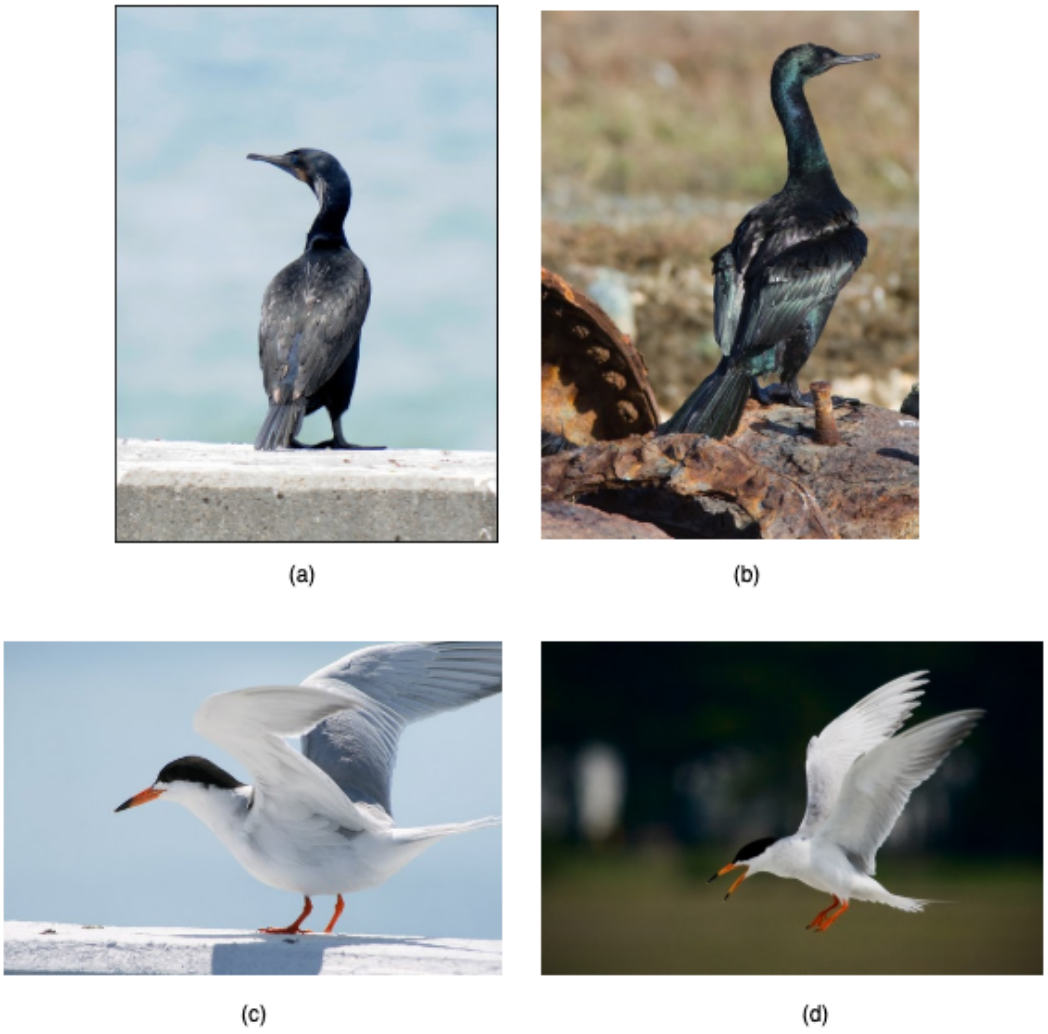


Figure 5.9: Bird pairs that are the hardest to distinguish during 50-way 5-shot classification. (a) *Brandt_Cormorant* (b) *Pelagic_Cormorant* (c) *Forsters_Tern* (d) *Common_Tern*

Moreover, this framework can be seen as an extension to a confusion matrix that can be generated using the prediction results in a classification task. A confusion matrix records the misclassifications between the predicted classes and visualises them in a way that emphasises which classes are often interchangeably predicted by the vision model. The proposed error analysis framework uses a measure of similarity to additionally capture the severity of the misclassifications of the vision model. The final results show an aggregated view of the confusion matrix in a quantitative manner.

I use this framework to evaluate all vision models trained according to the proposed approaches of this study during the subsequent experiments in few-shot image classification.

This chapter introduced the overall framework named ViOCE that is used to integrate ontology-based background knowledge into a DCNN-based vision model. Several instances of the vision model were informed by the different concept embeddings learnt in Chapter 4.4. The techniques used during training and inference with different embedding types were presented. The approaches were evaluated on tasks of few-shot image classification using all datasets presented in Chapter 4.

Results in 5-way classification with miniImageNet, tieredImageNet and Stanford Dogs datasets from Table 5.1 show that the model informed with n-ball concept embeddings performed better than others in all cases except during 5-way 1-shot classification. But during the20-way setting in Table 5.2, it performed the better than other even in the 1-shot case. This implies that the image embedding process with n-balls is harder than with point embeddings when the number of images and the number of candidate classes are very limited in a task. This observation is consistent with the results in Table 5.3 as well, where few-shot image classification is performed with fine-grained classes of the Stanford Dogs dataset.

Looking at the results of semantic meaningfulness of errors with the same datasets in Table 5.6, it can be seen how the meaningfulness of errors increase (i.e. the similarities between predicted and the ground truth class labels increase) with the model performance. This indicates that background knowledge helps to improve not only the performance, but also errors in a vision model.

In terms of the results in Table 5.4 for vision models informed by multi-relational n-ball embeddings, it can be seen that ViOCE (FO-EM) is performing better than the other models. But according to the earlier embedding quality results in Table 4.4, SP-EM produces better embeddings than FO-EM. I argue that this is due to FO-EM producing embeddings that are more favourable for the vision task when compared to SP-EM. The difference between FO-EM and SP-EM is the way object properties are handled during the embedding learning process. It can be concluded that representing them as translations on n-balls, in the case of SP-EM, is slightly ineffective when compared to eliminating translations and embedding only subsumptions and disjointnesses.

Looking at the error analysis from Section 5.9.2, again it is seen that highly similar classes are hard to classify for all models. Observing the example images in Figures 5.8 and 5.9, it is clear that these classes are visually very similar, resulting frequent classification errors.

Chapter 6

Conclusions

This chapter presents the overall conclusions of this thesis including the takeaways and the limitations of each contribution. Also, I present some future directions for the approaches proposed.

6.1 Thesis Overview

This thesis investigates the use of ontology-based background knowledge to inform a DCNN-based vision model when performing few-shot image classification. To this end, Chapter 4 addresses the first challenge of findings suitable ontologies that can be used as background knowledge for existing benchmark vision tasks. It describes how four OWL ontologies were constructed using knowledge sources such as WordNet and dataset annotations containing information such as the hierarchical structure and the visual features of the class labels of four image datasets. Chapter 4.4 presents the proposed approaches to learn ontology-based concept embeddings that can meaningfully represent each concept of an ontology in a vector space. Two main embedding types were investigated, namely, *CSim* and n-ball embeddings that can capture knowledge from an ontology in two different ways. I further propose techniques to determine the quality of the learnt embeddings. In Chapter 5.6, a framework was proposed to capture the semantic meaningfulness of the errors of a vision model and analyse them according to ontology-based background knowledge. The framework involves an ontology-based similarity measure that is selected according to the features of the ontology used. Subsequently, Chapter 5 presents the overall framework named ViOCE that is used for the integration of ontology-based concept embeddings with the vision model. The training and inference techniques used with each embedding type were

discussed and the performance results in several tasks of few-shot image classification are presented. Furthermore, the errors of each vision model were analysed for their semantic meaningfulness that explained the behaviour of the models with respect to the background knowledge used.

6.2 Contributions, Limitations and Future Directions

6.2.1 Ontology Construction for Image Datasets

I produced new OWL ontologies that extends the knowledge about the class labels found in the image datasets - miniImageNet, tieredImageNet, Stanford Dogs and CUB-200-2011.[1] These will contribute towards more research in the area of hybrid learning systems that combines vision tasks and ontologies.

Constructing the ontologies involved a few manual processes, hence it was a time consuming process. Hence, it will be beneficial to explore ways of automating the ontology construction process for a given dataset in the future. Moreover, the ontologies in this study can be further extended with more expert knowledge. For example, the miniImageNet ontology can be improved by including more features about the classes in their expressions, as it currently contains information only about the hierarchy of classes. In contrast, the birds ontology can benefit from more information about the hierarchy or a high level categorisation of the bird species, as it mostly contains only a set of attributes about a bird. I further identify the opportunity to improve the ontologies via the errors during the vision task. Misclassifications can point to missing information in the ontology that is used as background knowledge. These areas can be further investigated.

6.2.2 Learning Ontology-based Concept Embeddings

The concept embedding approaches in Chapter 4.4 were found to be effective in embedding concepts of an ontology in a vector space. It was shown how n-ball embeddings provide the opportunity to systematically determine the quality of the learnt embeddings with respect to the ontology-based knowledge. Whereas, *CSim* embeddings lack this capability.

[1] These ontologies are released via `https://github.com/miranthajayatilake/ViOCE-Ontologies`

Looking at the experimental results in Chapter 5 where the models informed with *n*-ball embeddings generally perform better others, it can be argued again that *n*-balls can represent the ontology information more faithfully than point form *CSim* embeddings. This is due to n-balls being able to utilise both centre and radius values, whereas *CSim* is limited to just points. For example, an *n*-ball enclosed by another represent the property of subsumption, whereas points represent subsumption only by the proximity of embeddings with arbitrary positioning in the vector space. Additionally, I present how n-balls provide the opportunity to quantitatively measure the quality of the embeddings learnt from an ontology.

A common limitation of both *CSim* and *n*-ball embeddings is that they do not perfectly model all knowledge from an ontology in the continuous space. The current *CSim* embedding approach only considers the specialisations (i.e., subsumption relations) of a concept and ignores other available knowledge such as partonomies. Going forward, this can be improved by incorporating more similarity information coming from the class expressions. With respect to *n*-ball embeddings, the loss function-driven learning process comes with an inherent loss of information. Additionally, I find that the proposed radii regularisation step can contribute to more loss of information if not handled carefully, as it limits the expressiveness of the *n*-balls in the vector space. This is further seen during the learning of multi-relational *n*-ball embeddings, where the translation of the *n*-balls representing the existential restrictions adds more noise to the vector space. Going forward, there are opportunities to investigate further the ways of controlling the *n*-balls with different axiom types, especially when embedding multi-relational information. Making use of other geometrical properties of the vector space and even considering different geometrical shapes when embedding will be interesting. This is to investigate how the expressiveness of the embeddings can be improved while understanding the computational costs involved.

Improvements to both *CSim* and *n*-ball embedding approaches are important, but I would encourage efforts to improve the *n*-ball embeddings when it comes to learning ontology-based concept embeddings.

6.2.3 Few-shot Image Classification informed by Concept Embeddings

The experimental results in Chapter 5 showed that overall the vision models informed with the concept embeddings produce superior performance in the tasks of few-shot

image classification compared to the exiting approaches. This demonstrates that the integration of ontology-based background knowledge can help a DCNN-based vision model to learn from few image examples.

When considering the few-shot classification performance generated by different types of embeddings used as background knowledge, it can be seen that *n*-ball embeddings generally performs better than others. One limitation observed during the training of the vision models with *n*-ball embeddings was the difficulty in projecting the image embeddings to be right inside the target *n*-balls, especially in the 1-shot case. This is also clear from the results where the models informed with *CSim* embeddings perform better during 5-way 1-shot tasks than the ones informed with *n*-ball embeddings. It was further noticed that if the candidate class embeddings were distantly placed from each other, this again made the mapping of image embeddings hard, resulting in lower classification accuracies.

Overall, the results show that an ontology can be a rich source of background knowledge for an informed machine learning approach performing few-shot image classification. The properties of ontology-based knowledge such as the well-defined structure, consistency and the ability to infer implicit information were found to be useful not only when learning with few examples, but also when improving the transparency of the predictions of a vision model with regard to its errors. Also, using ontology-based concept embeddings to guide the loss function during the training of a DCNN-based vision model is a good strategy when integrating the background knowledge.

Furthermore, I argue that all types of ontology information will contribute towards good concept embeddings. In the case of few-shot image classification, information from both the class hierarchy and other object properties in class expressions were found to be equally useful. But is noted that the extent of knowledge coming from a particular ontology feature can be important. For example, a shallow class hierarchy would not provide much use when learning the concept embeddings.

Going forward, improvements to the vision model design can be investigated in order to overcome the challenge of mapping image embeddings. Areas such as ensemble neural networks [ZWT02] and image transformers [RKH$^+$21] are interesting in this regard, given their recent successes. There is space to further investigate different vision tasks other than few-shot image classification, where ontology-based background knowledge can be used. Some prospects are tasks such as visual question answering [AAL$^+$15], image captioning [YJW$^+$16] and out-of-distribution detection [DT18].

Also, another interesting aspect would be enabling continuous integration of additional knowledge in the proposed framework. Background knowledge can be expanded and updated continuously over time. It will be useful to have techniques to iteratively update the downstream models that uses the background knowledge during applications.

6.2.4 Improved Error Analysis with Background Knowledge

The results of the proposed error analysis showed that in addition to better image classification accuracy, the informed vision models had better errors when looking at the semantic meaningfulness of the errors during the inference stage. It was presented how the frequently occurring errors were between classes that are highly similar which makes the classification harder. This method can be extended to any informed machine learning approach during evaluation in the future.

I further identify the opportunity to extend this analysis on semantic meaningfulness of errors towards approaches to explain the predictions of the vision model. Background knowledge is found to improve the transparency of the model's behaviour during inference and it will be important to address the aspect of explainability in the future.

Bibliography

[AAL+15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[ABK+07] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[ABRS20] Siddhant Arora, Srikanta Bedathur, Maya Ramanath, and Deepak Sharma. Iterefine: Iterative kg refinement embeddings using symbolic knowledge. *arXiv preprint arXiv:2006.04509*, 2020.

[ABV+20] Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, and Vince I Madai. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC Medical Informatics and Decision Making*, 20(1):1–9, 2020.

[ACIV17] Andrea Apicella, Anna Corazza, Francesco Isgro, and Giuseppe Vettigli. Integrating a priori probabilistic knowledge into classification for image description. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 197–199. IEEE, 2017.

[AJQS18] Muhammad Abdullah Jamal, Guo-Jun Qi, and Mubarak Shah. Task-agnostic meta-learning for few-shot learning. *arXiv e-prints*, pages arXiv–1805, 2018.

[APP+20] Saeed Amizadeh, Hamid Palangi, Alex Polozov, Yichen Huang, and Kazuhito Koishida. Neuro-symbolic visual reasoning: Disentangling.

In *International Conference on Machine Learning*, pages 279–290. PMLR, 2020.

[APS14] Tahani Alsubait, Bijan Parsia, and Uli Sattler. Measuring similarity in ontologies: a new family of measures. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 13–25. Springer, 2014.

[APS16] Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. Ontology-based multiple choice question generation. *KI-Künstliche Intelligenz*, 30(2):183–188, 2016.

[Asa19] Masataro Asai. Unsupervised grounding of plannable first-order logic representation from images. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 583–591, 2019.

[ASE17] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[ASST19] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International Conference on Machine Learning*, pages 232–241. PMLR, 2019.

[AWK+18] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. A survey of ontology learning techniques and applications. *Database*, 2018, 2018.

[BCR97] José Manuel Benítez, Juan Luis Castro, and Ignacio Requena. Are artificial neural networks black boxes? *IEEE Transactions on neural networks*, 8(5):1156–1164, 1997.

[BG18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[BGC+21] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[BGL14] Jiang Bian, Bin Gao, and Tie-Yan Liu. Knowledge-powered deep learning for word embedding. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 132–148. Springer, 2014.

[BGLL+20] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph kernels: State-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*, 2020.

[BH19] Federico Bianchi and Pascal Hitzler. On the capabilities of logic tensor networks for deductive reasoning. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2019.

[BHTV18a] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[BHTV18b] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[BJS17] Zied Bouraoui, Shoaib Jameel, and Steven Schockaert. Inductive reasoning about ontologies using conceptual spaces. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[BMR+20] Malik Boudiaf, Ziko Imtiaz Masud, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Transductive information maximization for few-shot learning. *arXiv preprint arXiv:2008.11297*, 2020.

[BMT17] Stephan Baier, Yunpu Ma, and Volker Tresp. Improving visual relationship detection using semantic modeling of scene descriptions. In *International Semantic Web Conference*, pages 53–68. Springer, 2017.

[BPL+16] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

[Bre08] Christopher A Brewster. Mind the gap: bridging from text to ontological knowledge. 2008.

[CFHP17] Davide Conigliaro, Roberta Ferrario, Céline Hudelot, and Daniele Porello. Integrating computer vision algorithms and ontologies for spectator crowd behavior analysis. In *Group and Crowd Behavior for Computer Vision*, pages 297–319. Elsevier, 2017.

[CFN16] Anthony Costa Constantinou, Norman Fenton, and Martin Neil. Integrating expert knowledge with data in bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved. *Expert systems with applications*, 56:197–208, 2016.

[CGG+20] Gabriele Ciravegna, Francesco Giannini, Marco Gori, Marco Maggini, and Stefano Melacci. Human-driven fol explanations of deep learning. In *IJCAI*, pages 2234–2240, 2020.

[CHJR+21] Jiaoyan Chen, Pan Hu, Ernesto Jimenez-Ruiz, Ole Magnus Holter, Denvar Antonyrajah, and Ian Horrocks. Owl2vec*: Embedding of owl ontologies. *Machine Learning*, pages 1–33, 2021.

[CLK+19a] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[CLK+19b] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[CLLC20] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Multi-scale adaptive task attention network for few-shot learning. *arXiv preprint arXiv:2011.14479*, 2020.

[CLLC21] Haoxing Chen, Huaxiong Li, Yaohui Li, and Chunlin Chen. Multi-level metric learning for few-shot image recognition. *arXiv preprint arXiv:2103.11383*, 2021.

[CMSV09] Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. Ontology learning. In *Handbook on ontologies*, pages 245–267. Springer, 2009.

[Cro12] Kevin Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the future of ict research. methods and approaches*, pages 210–221. Springer, 2012.

[CWB+11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[CWD+18] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[dBV19] M de Boer and JP Verhoosel. Creating and evaluating data-driven ontologies. *to appear*, 2019.

[DCRS19a] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[DCRS19b] Guneet S Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

[DDS+09a] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[DDS+09b] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[DGS16] Michelangelo Diligenti, Marco Gori, and Vincenzo Scoca. Learning efficiently in semantic based regularization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 33–46. Springer, 2016.

[DGS17] Michelangelo Diligenti, Marco Gori, and Claudio Sacca. Semantic-based regularization for learning and inference. *Artificial Intelligence*, 244:143–165, 2017.

[DLAG+20] Luca Di Liello, Pierfrancesco Ardino, Jacopo Gobbi, Paolo Morettin, Stefano Teso, and Andrea Passerini. Efficient generation of structured objects with constrained adversarial networks. *Advances in neural information processing systems*, 33, 2020.

[DLM20] Fabrizio Detassis, Michele Lombardi, and Michela Milano. Teaching the old dog new tricks: supervised learning with constraints. In *NeHuAI@ ECAI*, pages 44–51, 2020.

[DM15] Ernest Davis and Gary Marcus. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103, 2015.

[DRG17] Michelangelo Diligenti, Soumali Roychowdhury, and Marco Gori. Integrating prior knowledge into deep learning. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 920–923. IEEE, 2017.

[DRMD+19] Luc De Raedt, Robin Manhaeve, Sebastijan Dumancic, Thomas Demeester, and Angelika Kimmig. Neuro-symbolic= neural+ logical+ probabilistic. In *NeSy'19@ IJCAI, the 14th International Workshop on Neural-Symbolic Learning and Reasoning*, 2019.

[DRR16] Thomas Demeester, Tim Rocktäschel, and Sebastian Riedel. Lifted rule injection for relation embeddings. *arXiv preprint arXiv:1606.08359*, 2016.

[dSAdOSS18] Thamiris de Souza Alves, Caterine Silva de Oliveira, Cesar Sanin, and Edward Szczerbicki. From knowledge based vision systems to cognitive vision systems: a review. *Procedia Computer Science*, 126:1855–1864, 2018.

[DSG17] Ivan Donadello, Luciano Serafini, and Artur D'Avila Garcez. Logic tensor networks for semantic image interpretation. *arXiv preprint arXiv:1705.08968*, 2017.

[DSM19] Nikita Dvornik, Cordelia Schmid, and Julien Mairal. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3723–3731, 2019.

[DT18] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[EHSES+21] Noha A El-Hag, Ahmed Sedik, Walid El-Shafai, Heba M El-Hoseny, Ashraf AM Khalaf, Adel S El-Fishawy, Waleed Al-Nuaimy, Fathi E Abd El-Samie, and Ghada M El-Banby. Classification of retinal images based on convolutional neural network. *Microscopy Research and Technique*, 84(3):394–414, 2021.

[ELBS+15] Stefano Ermon, Ronan Le Bras, Santosh Suram, John Gregoire, Carla Gomes, Bart Selman, and Robert van Dover. Pattern decomposition with complex combinatorial constraints: Application to materials discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[ESB+18] Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie, Derek Doran, and Pascal Hitzler. Reasoning over rdf knowledge bases using deep learning. *arXiv preprint arXiv:1811.04132*, 2018.

[EW16] Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2, 2016.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[FCS+13] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013.

[FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37, 1996.

[FSCO18] Pierre Fournier, Olivier Sigaud, Mohamed Chetouani, and Pierre-Yves Oudeyer. Accuracy-based curriculum learning in deep reinforcement learning. *arXiv preprint arXiv:1806.09614*, 2018.

[FZD+19] Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. 2019.

[GAS16] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

[GB17] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[GCC+21] Yuxia Geng, Jiaoyan Chen, Zhuo Chen, Jeff Z Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen. Ontozsl: Ontology-enhanced zero-shot learning. In *Proceedings of the Web Conference 2021*, pages 3325–3336, 2021.

[Gei06] Peter Geibel. Reinforcement learning for mdps with constraints. In *European Conference on Machine Learning*, pages 646–653. Springer, 2006.

[GK18a] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[GK18b] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018.

[GSS14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Ham20] William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

[HBC+21] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37, 2021.

[HG17] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[HGP20a] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Exploiting unsupervised inputs for accurate few-shot classification. *arXiv e-prints*, pages arXiv–2001, 2020.

[HGP20b] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *arXiv preprint arXiv:2006.03806*, 2020.

[HGP20c] Yuqing Hu, Vincent Gripon, and Stéphane Pateux. Leveraging the feature distribution in transfer-based few-shot learning. *arXiv preprint arXiv:2006.03806*, 2020.

[HHC+19] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8460–8469, 2019.

[HL17] Patrick Hohenecker and Thomas Lukasiewicz. Deep learning for ontology reasoning. *arXiv preprint arXiv:1705.10342*, 2017.

[HL20] Patrick Hohenecker and Thomas Lukasiewicz. Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research*, 68:503–540, 2020.

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[HMCJR19a] Ole Magnus Holter, Erik Bryhn Myklebust, Jiaoyan Chen, and Ernesto Jiménez-Ruiz. Embedding owl ontologies with owl2vec. In *CEUR Workshop Proceedings*, volume 2456, pages 33–36. Technical University of Aachen, 2019.

[HMCJR19b] Ole Magnus Holter, Erik Bryhn Myklebust, Jiaoyan Chen, and Ernesto Jiménez-Ruiz. Embedding owl ontologies with owl2vec. In *CEUR Workshop Proceedings*, volume 2456, pages 33–36. Technical University of Aachen, 2019.

[HML+16] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv preprint arXiv:1603.06318*, 2016.

[HQDN19] Xianhao He, Peng Qiao, Yong Dou, and Xin Niu. Spatial attention network for few-shot learning. In *International Conference on Artificial Neural Networks*, pages 567–578. Springer, 2019.

[HYL17] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[HZRS16c] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.

[IHM+16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[IYIM20] León Illanes, Xi Yan, Rodrigo Toro Icarte, and Sheila A McIlraith. Symbolic plans as high-level instructions for reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 540–550, 2020.

[Jay19] Mirantha Jayathilaka. Enhancing generalization of first-order meta-learning. 2019.

[Ji19a] Qiang Ji. Combining knowledge with data for efficient and generalizable visual learning. *Pattern Recognition Letters*, 124:31–38, 2019.

[Ji19b] Qiang Ji. Combining knowledge with data for efficient and generalizable visual learning. *Pattern Recognition Letters*, 124:31–38, 2019.

[JMS20] Mirantha Jayathilaka, Tingting Mu, and Uli Sattler. Visual-semantic embedding model informed by structured knowledge. *arXiv preprint arXiv:2009.10026*, 2020.

[JMS21a] Mirantha Jayathilaka, Tingting Mu, and Uli Sattler. Ontology-based n-ball concept embeddings informing few-shot image classification. *arXiv preprint arXiv:2109.09063*, 2021.

[JMS21b] Mirantha Jayathilaka, Tingting Mu, and Uli Sattler. Towards knowledge-aware few-shot learning with ontology-based n-ball concept embeddings. *DOI: 10.13140/RG.2.2.26512.53769*, 2021.

[JXLL18a] Chenhan Jiang, Hang Xu, Xiangdan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. *arXiv preprint arXiv:1810.12681*, 2018.

[JXLL18b] Chenhan Jiang, Hang Xu, Xiangdan Liang, and Liang Lin. Hybrid knowledge routed modules for large-scale object detection. *arXiv preprint arXiv:1810.12681*, 2018.

[JZSS16] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016.

[KBB+12] Angelika Kimmig, Stephen Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, pages 1–4, 2012.

[KC10] Stasinos Konstantopoulos and Angelos Charalambidis. Formulating description logic learning as an inductive logic programming task. In *International Conference on Fuzzy Systems*, pages 1–7. IEEE, 2010.

[KFR06] Vladimir M Krasnopolsky and Michael S Fox-Rabinovitz. Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. *Neural Networks*, 19(2):122–134, 2006.

[KHTT+16] Reinier Kop, Mark Hoogendoorn, Annette Ten Teije, Frederike L Büchner, Pauline Slottje, Leon MG Moons, and Mattijs E Numans. Predictive modeling of colorectal cancer using a dedicated preprocessing pipeline on routine electronic medical records. *Computers in biology and medicine*, 76:30–38, 2016.

[KJYFF11] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[KLWYH19] Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. El embeddings: Geometric construction of models for the description logic el++. *arXiv preprint arXiv:1902.10499*, 2019.

[KSH12a] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[KSH12b] 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.

[KSKW15] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[KWRK17] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[KZG+16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[KZS15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[LBL18] Joao Loula, Marco Baroni, and Brenden M Lake. Rearranging the familiar: Testing compositional generalization in recurrent networks. *arXiv preprint arXiv:1807.07545*, 2018.

[Leo08] Sabina Leonelli. Bio-ontologies as tools for integration in biology. 2008.

[LG14] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.

[LGF16a] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *International conference on machine learning*, pages 430–438. PMLR, 2016.

[LGF16b] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. In *International conference on machine learning*, pages 430–438. PMLR, 2016.

[LGG+20] Luis C Lamb, Artur Garcez, Marco Gori, Marcelo Prates, Pedro Avelar, and Moshe Vardi. Graph neural networks meet neural-symbolic computing: A survey and perspective. *arXiv preprint arXiv:2003.00330*, 2020.

[LHZ+18] Xiaodan Liang, Zhiting Hu, Hao Zhang, Liang Lin, and Eric P Xing. Symbolic graph reasoning meets convolutions. *Advances in Neural Information Processing Systems*, 31:1853–1863, 2018.

[LLX+19] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9715–9724, 2019.

[LMRS19] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[LTHS88] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28, 1988.

[LUTG17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[LWX+19] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[LY18] Yang Li and Tao Yang. Word embedding for understanding natural language: a survey. In *Guide to big data applications*, pages 83–104. Springer, 2018.

[LZCL17] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[Mar18] Gary Marcus. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*, 2018.

[MCCD13a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MCCD13b] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MCSFL15] Yi Ma, Paul A Crook, Ruhi Sarikaya, and Eric Fosler-Lussier. Knowledge graph inference for spoken dialog systems. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5346–5350. IEEE, 2015.

[MCX18] Tengfei Ma, Jie Chen, and Cao Xiao. Constrained generation of semantically valid graphs via regularizing variational autoencoders. *arXiv preprint arXiv:1809.02630*, 2018.

[MDG+20] Giuseppe Marra, Michelangelo Diligenti, Francesco Giannini, Marco Gori, and Marco Maggini. Relational neural machines. *arXiv preprint arXiv:2002.02193*, 2020.

[MDK+18] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *arXiv preprint arXiv:1805.10872*, 2018.

[MGDG19] Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Lyrics: A general interface layer to integrate logic inference and deep learning. *arXiv preprint arXiv:1903.07534*, 2019.

[MGTA12] Tingting Mu, John Yannis Goulermas, Jun'ichi Tsujii, and Sophia Ananiadou. Proximity-based frameworks for generating embeddings from multi-output data. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2216–2232, 2012.

[MH80] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[Mil95] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[MJWG12] Tingting Mu, Jianmin Jiang, Yan Wang, and John Y Goulermas. Adaptive data embedding framework for multiclass classification. *IEEE transactions on neural networks and learning systems*, 23(8):1291–1303, 2012.

[MKS+20] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[MPSP+09a] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.

[MPSP+09b] Boris Motik, Peter F Patel-Schneider, Bijan Parsia, Conrad Bock, Achille Fokoue, Peter Haase, Rinke Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, et al. Owl 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation*, 27(65):159, 2009.

[MRCA17] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[MSG16a] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016.

[MSG16b] Kenneth Marino, Ruslan Salakhutdinov, and Abhinav Gupta. The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*, 2016.

[MVRV99] Gary F Marcus, Sugumaran Vijayan, S Bandi Rao, and Peter M Vishton. Rule learning by seven-month-old infants. *Science*, 283(5398):77–80, 1999.

[MXH+21] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10573–10582, 2021.

[MY17] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[MYMT18] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018.

[N+04] Richard E Neapolitan et al. *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[NAS18] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[NDL+05] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, 2005.

[NMB+13] Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S Corrado, and Jeffrey Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[NMTG15] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2015.

[ORL18] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

[Pan06] Jeff Z Pan. A flexible ontology reasoning architecture for the semantic web. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):246–260, 2006.

[Pau17] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

[PHBB16] Eunbyung Park, Xufeng Han, Tamara L Berg, and Alexander C Berg. Combining multiple sources of knowledge in deep cnns for action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[PLZ+19] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 441–449, 2019.

[PMPS18] Ankur Padia, David Martin, and Peter F Patel-Schneider. Automating class/instance representational choices in knowledge bases. In *European Knowledge Acquisition Workshop*, pages 273–288. Springer, 2018.

[PY09] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[PZL+18] Julius Pfrommer, Clemens Zimmerling, Jinzhao Liu, Luise Kärger, Frank Henning, and Jürgen Beyerer. Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CiRP*, 72:426–431, 2018.

[QBL18] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.

[QSL+19] Limeng Qiao, Yemin Shi, Jia Li, Yaowei Wang, Tiejun Huang, and Yonghong Tian. Transductive episodic-wise adaptive metric for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3603–3612, 2019.

[QYLC18] Zhuwei Qin, Fuxun Yu, Chenchen Liu, and Xiang Chen. How convolutional neural network see the world-a survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191*, 2018.

[RBS19] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 331–339, 2019.

[RD06] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[RDGZ+20] Felipe Riquelme, Alfredo De Goyeneche, Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. Explaining vqa predictions using visual grounding and a knowledge base. *Image and Vision Computing*, 101:103968, 2020.

[RGH18] Miguel Ángel Rodríguez-García and Robert Hoehndorf. Inferring ontology graph structures using owl reasoning. *BMC bioinformatics*, 19(1):1–9, 2018.

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[RL16a] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[RL16b] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[RRS+18a] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[RRS+18b] Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*, 2018.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[RSR15] Tim Rocktäschel, Sameer Singh, and Sebastian Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the 2015 conference of the north American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1119–1129, 2015.

[RSS+18] Hongyu Ren, Russell Stewart, Jiaming Song, Volodymyr Kuleshov, and Stefano Ermon. Adversarial constraint learning for structured prediction. *arXiv preprint arXiv:1805.10561*, 2018.

[RTR+18] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[SBRS18] Chakaveh Saedi, António Branco, João Rodrigues, and Joao Silva. Wordnet embeddings. In *Proceedings of the third workshop on representation learning for NLP*, pages 122–131, 2018.

[SC18] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond

accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018.

[SDG17] Luciano Serafini, Ivan Donadello, and Artur d'Avila Garcez. Learning and reasoning in logic tensor networks: theory and application to semantic image interpretation. In *Proceedings of the Symposium on Applied Computing*, pages 125–130. ACM, 2017.

[SDSGR18] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018.

[SFH17] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

[SG16] Luciano Serafini and Artur d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. *arXiv preprint arXiv:1606.04422*, 2016.

[SGMN13] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, 26:935–943, 2013.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[SHS+18] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[Sin12] Amit Singhal. Introducing the knowledge graph: things, not strings, May 2012.

[SKCL13] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature

learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3626–3633, 2013.

[SKM+13] Tara N Sainath, Brian Kingsbury, Abdel-rahman Mohamed, George E Dahl, George Saon, Hagen Soltau, Tomas Beran, Aleksandr Y Aravkin, and Bhuvana Ramabhadran. Improvements to deep convolutional neural networks for lvcsr. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 315–320. IEEE, 2013.

[SLCS19] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[SLM21] Mattia Silvestri, Michele Lombardi, and Michela Milano. Injecting domain knowledge in neural networks: a controlled experiment on a constrained problem. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 266–282. Springer, 2021.

[SM13] Daniel Soutner and Luděk Müller. Application of lstm neural networks in language modelling. In *International Conference on Text, Speech and Dialogue*, pages 105–112. Springer, 2013.

[SMC12] Jurgen Schmidhuber, U Meier, and D Ciresan. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649. IEEE Computer Society, 2012.

[SMH08] Rob Shearer, Boris Motik, and Ian Horrocks. Hermit: A highly-efficient owl reasoner. In *Owled*, volume 432, page 91, 2008.

[SOPH16] Fabio Alexandre Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. Breast cancer histopathological image classification using convolutional neural networks. In *2016 international joint conference on neural networks (IJCNN)*, pages 2560–2567. IEEE, 2016.

[SSS+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker,

Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[SSZ17] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[SWM17] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.

[SXD+17] Md Kamruzzaman Sarker, Ning Xie, Derek Doran, Michael Raymer, and Pascal Hitzler. Explaining trained neural networks with semantic web technologies: First steps. *arXiv preprint arXiv:1710.04324*, 2017.

[SYZ+18a] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[SYZ+18b] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TdM15] Ilaria Tiddi, Mathieu d'Aquin, and Enrico Motta. Data patterns explained with linked data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 271–275. Springer, 2015.

[TGJ+15] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015.

[TS94] Geoffrey G Towell and Jude W Shavlik. Knowledge-based artificial neural networks. *Artificial intelligence*, 70(1-2):119–165, 1994.

[TS10] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[TS14] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[TWK+20] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[TYRW14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[vBdBvH+21] Michael van Bekkum, Maaike de Boer, Frank van Harmelen, André Meyer-Vitali, and Annette ten Teije. Modular design patterns for hybrid learning and reasoning systems. *Applied Intelligence*, pages 1–19, 2021.

[VBL+16a] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

[VBL+16b] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[VDDP18] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[VdMH08] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[VRMG+19] Laura Von Rueden, Sebastian Mayer, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. Informed machine learning–towards a taxonomy of explicit integration of knowledge into machine learning. *learning*, 18:19–20, 2019.

[VTBE15] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[VZ11] Luc Vinet and Alexei Zhedanov. A 'missing'family of classical orthogonal polynomials. *Journal of Physics A: Mathematical and Theoretical*, 44(8):085201, 2011.

[WB18] Daniel S Weld and Gagan Bansal. Intelligible artificial intelligence. *ArXiv e-prints, March 2018*, 2018.

[WBW+11a] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[WBW+11b] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[WGHH18] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286, 2018.

[WLB12] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):1–36, 2012.

[WLGJ19] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. Parn: Position-aware relation networks for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6659–6667, 2019.

[WMWG17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[WNS+11] Patricia L Whetzel, Natalya F Noy, Nigam H Shah, Paul R Alexander, Csongor Nyulas, Tania Tudorache, and Mark A Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl_2):W541–W545, 2011.

[WWX17] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2(3):034603, 2017.

[WXC04] Lei Wang, Ping Xue, and Kap Luk Chan. Incorporating prior knowledge into svm for image retrieval. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 981–984. IEEE, 2004.

[WYG18a] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6857–6866, 2018.

[WYG18b] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6857–6866, 2018.

[WYM+16] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[XFM+19] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd international*

ACM SIGIR conference on research and development in information retrieval, pages 285–294, 2019.

[XXY+15] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

[XZF+18] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning*, pages 5502–5511. PMLR, 2018.

[YHZL19] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

[YJW+16] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[ZPW+19] Wen Zhang, Bibek Paudel, Liang Wang, Jiaoyan Chen, Hai Zhu, Wei Zhang, Abraham Bernstein, and Huajun Chen. Iteratively learning embeddings and rules for knowledge graph reasoning. In *The World Wide Web Conference*, pages 2366–2377, 2019.

[ZQS+18] Xueting Zhang, Yuting Qiang, Flood Sung, Yongxin Yang, and Timothy M Hospedales. Relationnet2: Deep comparison columns for few-shot learning. *arXiv preprint arXiv:1811.07100*, 2018.

[ZWT02] Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

[ZZN+19] Jian Zhang, Chenglong Zhao, Bingbing Ni, Minghao Xu, and Xiaokang Yang. Variational few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1685–1694, 2019.

Appendix A

Appendix

A.1 Few-shot Image Classification Settings

Table A.1 lists different settings that few-shot image classification is evaluated in the existing approaches.

Table A.1: List of exiting datasets and evaluation settings in few-shot image classification

Paper	Datasets	Evaluation metics
Qiao et al.	Full ImageNet (ILSVRC2015)	1000-way / 1, 2, 3-shot
He et al.	CUB-200-2011 Stanford Dogs	5-way / 1, 5-shot
Vinyals et al.	Omniglot	5-way / 1, 5-shot 20-way / 1, 5-shot
	Full ImageNet (ILSVRC2015)	5-way / 1-shot
Chen et al.	CUB-200-2011	5-way / 1, 5-shot
Zhang et al.	CUB-200-2011 Flower102	5-way / 1, 5-shot
Sun et al.	CUB-200-2011	100-way / 1, 2, 5-shot
	miniPPlankton Stanford Dogs	5-way / 1, 5-shot
Xu et al.	Full ImageNet (ILSVRC2015)	100-way / 1, 5, 10-shot
	OpenImages	100-way / 5-shot

Sung et al.	Omniglot	5-way / 1, 5-shot 20-way / 1, 5-shot
Wang et al.	Never-Ending Image Learning (NEIL) Full ImageNet (ILSVRC2015)	Zero shot 2-hop/3-hop
Triantafillou et al.	Omniglot miniImageNet CUB-200-2011	5,20-way / 1,5-shot 5,20-way / 1-shot (image retrieval) 5,20-way / 1-shot (image retrieval)
Hilliard et al.	CUB-200-2011	5-way / 1, 5-shot
Zhu et al.	CUB-200-2011 Stanford Dogs Stanford Cars FGVC Aircraft	5-way / 1, 5-shot
Hu et al.	tieredImageNet CUB-200-2011 CIFAR-FS	5-way / 1, 5-shot
Chen et al.	CUB-200-2011	5-way / 1, 5-shot
Li et al.	Omniglot	5-way / 1, 5-shot
Park et al.	tieredImagenet Omniglot	5-way / 1, 5-shot