Understanding Deep Representations in CNNs from Concepts to Relations to Rules

Eric Ferreira dos Santos M.Sc.

Supervised by Prof. Alessandra Mileo

A thesis presented for the degree of Doctor of Philosophy (Ph.D.)

School of Computing

Dublin City University

January 2025

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Eric Ferreira dos Santos

ID No.: 20215568

30/01/2025

Acknowledgements

First, I would like to thank God for conducting me throughout the journey, giving me health and knowledge, and blessing me with life. I could not achieve this milestone without Him.

I would like to express my sincere gratitude to my supervisor, Prof. Alessandra Mileo, for her invaluable advice, expertise and support throughout this process, and Prof. Suzanne Little, Janet Choi, Angela Lally and Vicky Flanagan for all support and help.

I would also like to thank my wife, Veronica, for her love and support during the challenging times of moving abroad and adapting to a new environment, and also my mother, Rosana and grandmother, Darcy, for their continuous encouragement and belief in me, which contributed enormously to each achievement during my PhD journey.

Finally, I would like to express my appreciation to all those who have directly or indirectly contributed to this work, including my colleagues and fellow researchers. Thank you for your support and collaboration.

Contents

List of Figures

List of Tables

Abstract

Understanding Deep Representations in CNNs from Concepts to Relations to Rules

Eric Ferreira dos Santos

The field of Explainable Artificial Intelligence (*XAI*) has recently gained prominence, driven by the demands for transparency and accountability in the application of *AI* models in critical decision-making scenarios. Although Deep Neural Networks (*DNNs*) have achieved remarkable success, specifically in computer vision tasks, understanding the steps pertaining to their decision-making processes remains a significant hurdle due to the opaque nature of the model. Most existing methods for explanation in computer vision focus on low-level features (such as pixels) and their influence on the final classification. This relevance is then used to generate explanations based on visual cues (such as saliency maps), rather than providing higher-level human concepts and the relations among them. This thesis aims to address this issue by proposing approaches to extract high-level human concepts, relations and rules from deep representations, thus enhancing the transparency of the model's decision-making process. Specifically, we focus on Convolutional Neural Networks (*CNNs*) for image classification tasks and have leveraged our approach to make assumptions about what semantic concepts and relations are effectively learned and hidden in deep representations for computer vision models. Our results demonstrate the approach's ability to provide a semantic understanding of what a trained image classification model has learned in a way that humans can comprehend without a deeper knowledge of machine learning techniques and concepts, thereby providing transparency and acceptance of the model's results.

Chapter 1

Introduction

Explainable Artificial Intelligence (XAI) is a field of research dedicated to increasing transparency in AI system decision-making processes [46]. With the proliferation of Deep Neural Networks ($DNNs$) and the push to use these models in critical decision-making scenarios, XAI researchers are emphasising the importance of explaining the reasoning behind DNN decisions [13]. Such clarifications are essential to ensure the models' dependability [120] and to adhere to the European Union's latest AI regulations[1]. This is particularly essential in high-stake decision-making, where the price to pay for wrong decisions is high, and therefore, understanding and explaining the rationale behind a given outcome is paramount. This includes scenarios such as medical diagnostics, law enforcement, and financial analysis. Despite the urgency to enhance modern deep learning approaches with explainability, due to the black-box nature of these models, it has proven challenging to significantly enhance transparency and human understanding of DNN decisions beyond visual cues.

Recently, there has been a significant increase in the number of published research papers focusing on interpretability [43], with various approaches formulated to provide an interpretation of the elements influencing the decision-making in $DNNs$. Although significant progress has been made in this area, most of the interpretability methods in Computer Vision still rely heavily on training data and their low-level

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689

features [81, 104]. As a result, these techniques often fail to incorporate human domain expertise in their explanations, resulting in a lack of clarity and transparency from the perspective of human expert knowledge.

Alongside the need for generating explanations, it is essential that an explanation itself is presented in a form that a human can understand. In other words, the additional information provided on the *DNN* decision-making process should make the output produced by the model *self-explainable* for a human expert, without the need for human interpretation, which may, in turn, be prone to additional noise such as human bias and human error.

According to [102], in order to trust a model's choice, the explanation must present the following characteristics:

- Clearly Interpretable: Explanations need to be clear and easy to understand. Users should grasp the rationale behind a model's decisions without needing deep technical expertise.

- Relatable to the User: Explanations must link to the user's experiences and context. When the model's decisions are related to familiar concepts or scenarios, users can more effectively understand its operation and the reasons behind its conclusions.

- Tying Decisions to Contextual Information: Adequate explanations should connect decisions with pertinent contextual details regarding the options being considered. This requires incorporating information about the situation or environment surrounding the decision.

- Demonstrating Intermediate Thinking: Users must be able to see the model's intermediate reasoning. By showing the steps or considerations that contribute to a final decision, users can understand how the model processes information.

This thesis proposes a framework to provide a transparent and reliable explanation of the inner workings of a deep neural model. This will pave the way for users to understand and interact with the model clearly and in such a way that they

do not need knowledge of the technology but can interpret it with their common sense knowledge. The framework result is an explanation aligned to three of the characteristics provided by [102], namely clearly interpretable, relatable to the user and presenting the model's intermediate thinking.

The design focuses on Computer Vision tasks, a branch of Artificial Intelligence that looks into how computer programs can interpret, represent, and act on visual inputs (such as pictures and videos). Deep Learning models such as Convolutional Neural Networks (*CNNs*) are tailored explicitly to computer vision tasks. In the last decade, they have become remarkably successful and popular, achieving incredible accuracy in image classification and object detection tasks.

To enable a better understanding of deep representations in human terms, the framework proposed in this thesis has the following characteristics: 1) the ability to look into a deep *CNN* representation to identify what semantic concepts have been learned that affect the network output in a specific classification task, 2) the ability to relate concepts with each other using external knowledge (via Knowledge Graphs), and identify what part of such relational knowledge is likely to have been learned by the *DNN* and used for the specific task, and 3) the ability to use semantic understanding as concepts and relations to induce generalised logical rules that can approximate the *DNN* decision making process in a declarative and explainable representation space.

This Chapter introduces foundational knowledge and background on key areas that are relevant for the remainder of the thesis. The following sections cover: the areas of Explainable *AI*, presenting the general objectives of the field as applied to *DNNs*; an introduction to Convolutional Neural Networks (*CNNs*), a type of deep neural network used in computer vision tasks, from which the primary representations will be extracted in this research; the concepts around Knowledge Graphs, employed as an external knowledge to enhance the transparency of the learned representations; and an overview of Rule Learning, a process used in the third component of the framework to extract deductive rules representing the mechanism behind the

deep learning process.

1.1 Introduction to Explainable AI

Deep neural networks (*DNNs*) are well-established approaches demonstrating significant performance in complex tasks such as computer vision and language understanding. Despite their success, limitations hinder their wider adoption, including limited ability to generalise to other domains and datasets, high dependency on data availability, lack of transparency and explainability [124].

The interest in making the inner workings of *DNN* explicit in human terms is not new. Efforts to understand *DNNs* were made even when insufficient data quality and availability and limited computational resources prevented the full utilisation of deep learning technologies [122, 11, 83]. Due to neural networks' opaque nature and data complexity, providing an explanation that a machine learning non-specialised user can understand and interact with is not trivial. This difficulty hinders using *DNNs* in critical environments such as diagnostic imaging, disaster management and security surveillance. In these scenarios, it is essential to understand why the model generated a given outcome, specifically what internal representation the model has learned from the training data and how this representation contributed to such an outcome. The ability to do this can be instrumental in identifying and correcting mistakes (debugging) and detecting and eventually mitigating potential bias in the data or the model.

An example in computer vision is some methods that rely on visual clues about the model's output [109], which do not clarify the semantic concept learned by the model specifically. On the other hand, some methods implement a link between a concept representation and the model's output [14] but without providing a reasonable relation between the concepts. This highlights the complexity of generating an explanation of what the model learned.

The Explainable Artificial Intelligence (*XAI*) field is concerned with investigating methods and techniques aimed at clarifying the decisions made by an *AI* artefact,

allowing users to interact with and evaluate the outcomes. Specifically for computer vision tasks, *XAI's* methods can be categorised based on the incorporation stage, the explanation scope, the aim of the process, the explanation modality, and the training distribution [59]. The incorporation stage has two primary separations: ante-hoc and post-hoc, which relate to which part of the process an explanation method will be applied to. Ante-hoc techniques can change the model structure during the learning process to achieve an interpretation. Post-hoc approaches, on the other hand, aim to explore the outcomes of a trained model without changing its structure.

The scope of explanation can vary, ranging from explaining individual predictions to describing the overall behaviour of the model. When we discuss a classification model, the *XAI* method can either focus on interpreting the outcome for a single instance, also known as a local explanation, or to understand the entire model, also known as a global explanation. The methods can explain the model output for a different purpose, either to explain the model's decision-making process or to improve the model's performance and robustness.

Regarding the explanation modality, the explanations provided by *XAI* methods can take different forms, such as visual heat maps, textual descriptions, counterfactual examples, or rules. The methods also might require a different approach to the model's training, either needing additional training data or annotations to learn the explanations, while others operate on the trained model without further training.

Current explainability methods still presents numerous open challenges [59] including (but not limited) to the following:

- generalizability of explainability methods across different neural network architectures;
- ability to explain models output using fewer examples;
- lack of robust evaluation metrics that can take into account human feedback;
- lack of benchmarks when it comes to assess the quality of explanations for

humans.

If we consider only the issue of evaluating explanations in *XAI*, for example, this is a significant challenge due to the lack of standard and robust evaluation protocols, along with the lack of consensus on what constitutes a "good" explanation. This issue is further complicated by the multidisciplinary nature of *XAI*, which introduces diverse perspectives and inconsistencies in evaluation methodologies across application domains [62].

Studies indicate that, although various metrics and frameworks have been proposed, there is still no agreement on which properties - such as fidelity, usability, or fairness - should be prioritised, nor on what constitutes an effective measurement [79]. In fact, user-centred evaluations are often inconsistent, with many relying on ad-hoc approaches that hinder reproducibility and cross-study comparisons [92]. Furthermore, the subjective nature of human-centred evaluations complicates efforts towards establishing general standards. The absence of a unified approach limits the ability to compare methods across studies and impedes progress in reconciling technical and human-centric goals in *XAI* evaluation.

This work proposes an approach to generate a comprehensible representation of what the model has learned for the end user, applicable to transfer learning models and capable of learning from fewer examples.

1.2 Introduction to Convolutional Neural Networks

Convolutional Neural Networks (*CNNs*) [74] are specialised deep neural networks crafted explicitly for handling structured grid data, such as images composed by pixels. The design of *CNNs* takes inspiration from the visual cortex of animals, which processes visual input through multiple layers that gradually extract more intricate features. *CNNs* use convolutional layers to implement filters that capture spatial hierarchies within the data, enabling the recognition of patterns such as edges, textures, and shapes. This layered feature extraction approach provides a

considerable advantage over traditional feed-forward neural networks by decreasing the number of parameters and computational load, thus making *CNNs* more effective for tasks involving images [90, 142].

The applications of *CNNs* are vast and varied, with significant impact in fields such as computer vision, natural language processing, and even audio recognition. In computer vision, *CNNs* have been successfully employed for tasks like image classification, object detection, and facial recognition. For instance, *CNNs* are integral to systems used in autonomous vehicles for recognising traffic signs and pedestrians, enhancing safety and navigation capabilities. In the medical field, *CNNs* can assist radiologists in the interpretation of medical images for diagnostics. Additionally, *CNNs* have been adapted for text classification and sentiment analysis, showcasing their versatility beyond image processing [90, 3].

Despite the growing popularity and efficiency of Transformers architectures - architectures focused on attention and specifically suited to learn statistical patterns in sequences such as text [125] - , particularly Vision Transformers (*ViTs*) in tasks like image processing, there are compelling reasons to continue utilising Convolutional Neural Networks (*CNNs*). One significant concern is the tendency of *ViTs* to amplify biases present in training data, as highlighted in [84], which discusses how *ViTs* may exacerbate gender bias due to their attention mechanisms. Additionally, *CNNs* maintain computational efficiency and performance advantages, especially on smaller datasets where *ViTs* struggle due to their lack of strong inductive biases for spatial relevance and diverse channel representation [80]. Moreover, *CNNs* excel at capturing high-frequency components crucial for tasks requiring detailed texture recognition, which is essential in fields like medical imaging, where precision is paramount [10]. Thus, while Transformers offer innovative approaches, the unique strengths of *CNNs* ensure they remain a vital component of the machine learning landscape.

Examples of the success of *CNNs* can be seen looking at the ImageNet Large

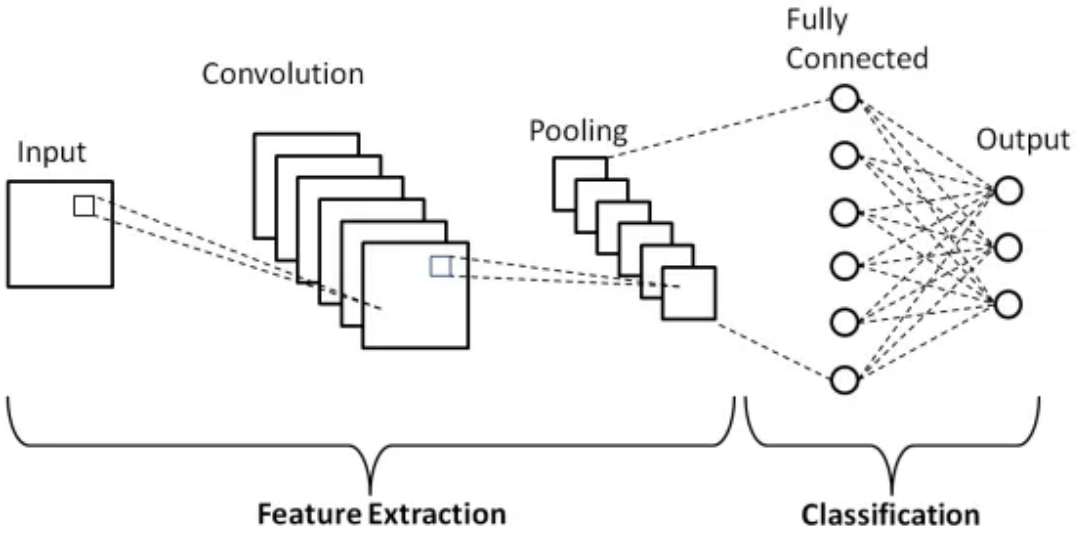

Figure 1.1: *CNN* architecture [12]

Scale Visual Recognition Challenge (*ILSVRC*)[2]. Models such as *AlexNet* [68], *VGGNet* [113], and *ResNet* [51] have shown exceptional accuracy in image classification tasks. Just over a decade ago, *AlexNet* significantly lowered the error rate compared to earlier methods, signifying a pivotal moment in deep learning and *CNN* research. These breakthroughs have resulted in the extensive adoption of *CNNs* in both academic studies and commercial uses, as they persist in expanding machine learning capabilities [142, 44].

CNN architectures are typically composed by three types of layers, namely convolutional, pooling, and fully connected layers (Figure 1.1). Convolutional layers use filters on the input data to generate feature maps, while pooling layers decrease the spatial dimensions of these maps, preserving essential information and lowering computational requirements. After a series of convolutional and pooling layers aimed at learning low level features, fully connected layers are task-specific and perform the final classification task. This multi-layered structure enables *CNNs* to learn intricate data representations, making them highly effective for tasks involving high-dimensional inputs like images and videos [3, 44].

The ongoing exploration of *CNNs* is geared towards enhancing their efficiency, robustness, and interpretability. Approaches like transfer learning, in which pre-

[2]https://image-net.org/challenges/LSVRC/

trained models are adapted to new tasks, have gained traction due to their ability to use existing knowledge and shorten training time. Moreover, progress in architectures, including the advent of residual networks (*ResNets*) [51, 140] and attention mechanisms, has further boosted the efficacy of *CNNs*. As deep learning continues to advance, *CNNs* remain at the cutting edge of research, fostering innovations in multiple fields and setting new standards for artificial intelligence [58].

1.3 Introduction to Knowledge Graphs

Knowledge Graphs (*KGs*) are structured representations of knowledge where entities are nodes, and relations between these entities are edges. *KGs* can integrate vast amounts of data from heterogeneous sources, offering a semantic representation that is both machine-readable and human-interpretable [144]. The use of *KGs* has emerged as promising for explainable *AI*, as they are intrinsically explainable [73, 119]. The structured nature of *KGs* allows for the generation of explanations in a natural, human-friendly format. By navigating the paths between entities in a *KG*, one can uncover the reasoning behind a model's prediction [126]. This is a desirable property that can be leveraged to address the "black box" nature of deep learning models.

While powerful tools for representing and integrating heterogeneous data, *KGs* face several key limitations, primarily related to their construction (it requires time to build it manually), scalability, data quality, and dynamic updates. As the volume of data grows exponentially, traditional methods of constructing and maintaining *KGs* struggle to keep pace, often leading to incomplete or outdated representations. However, advancements in automated methods, such as machine learning and natural language processing, have fuelled the creation of *KGs*. The availability of large-scale data sources and the development of algorithms for knowledge extraction and inference have enabled the creation of sophisticated and complete *KGs*, mitigating these challenges.

Constructing a *KG* automatically from unstructured data typically involves three

main steps: (i) entity recognition, (ii) relation extraction, and (iii) linking and integration. In the first step, all the entities within the unstructured data need to be identified, creating the graph nodes. The second step is responsible for identifying the relations, where edges between the nodes (representing the relation between the entities) are created. Last, the entities and relations can be associated with existing knowledge bases to improve coverage and consistency. Sophisticated methods, such as *BertNet* [50], which uses pre-trained language models to improve the *KG's* efficiency and accuracy, and Retrieval-Augmented Generation (*RAG*) [35], applied to extract relevant information from external knowledge to create a *KG*, have increased the *KG's* applicability. They have been widely used in Natural Language Processing, including question answering, information extraction, and entity recognition [23]. They have also been used to combine data from numerous sources, making silos data more accessible for querying and inference.

Early *KGs*, such as *Freebase* [17] and *DBpedia* [7] laid the groundwork by demonstrating the utility of *KGs* in information retrieval and search engines. For instance, in the recommendation systems space, *KGs* can be used to explain why a particular item is suggested based on the relations between users, items, and contextual information [139]. In the healthcare domain, authors of [117] developed rules in a medical *KG* to assess the clinical rationality of medical claims and to identify the suspected claims by reasoning. *KGs* have been applied in different studies to perform reasoning and find missing information in tax refunds [132], as well as to extract structured events from financial news and to provide external knowledge that can be used to embed events and forecast stock trends [32]. There are other areas which use knowledge graphs to increase model transparency [100], which highlights their potential role in enhancing machine learning model's interpretability.

1.4 Introduction to Inductive Rule Learning

Rule learning is a significant area within artificial intelligence that focuses on generating rules from data, existing rules or models [129]. Rule learning involves three

kinds of inferences: inductive, deductive, and analogical. Such rules are meant to generalise the patterns in data and can be used for classification and prediction tasks. At its core, inductive rule learning (or rule induction) involves the process of characterising, by induction, a set of rules or a decision list that can effectively categorise or predict outcomes based on input features.

While conventional machine learning algorithms aim to uncover latent patterns from large datasets, rule-based *AI* algorithms utilise logical reasoning to manipulate symbols and perform symbolic inference. The most common approach in inductive rule learning is the separate-and-conquer method [39], which learns one rule at a time and iteratively removes examples covered by the learned rules. Algorithms, such as *PROGOL* [91] and *FOIL* [97], are examples of the approaches used.

A key disadvantage of inductive rule learning is the risk of overfitting when a set of rules becomes too specific to the data, leading to poor generalisation. Another issue is the massive amount of available data, where the search space to generate such rules becomes impracticable, as the effort to generate it manually. To overcome these problems, logical programs are used to learn the rules that describe a dataset. Inductive Logical Programming (*ILP*), a subfield of inductive rule learning, aims to learn a hypothesis (a set of rules), which combined with a background knowledge base, can explain a set of observations [69].

ILP is motivated by its use of logic as a declarative representation, making hypotheses easily understandable and interpretable. Through logic, inductive logic programming systems can also utilise background knowledge during the induction process. This background knowledge may include definitions of auxiliary relations or predicates that the learner can apply [99]. The use of background knowledge can improve the generalisation. The applications of *ILP* are diverse and span across several domains, such as cybersecurity [45], image classification [6, 24], event detection [70], natural language processing [72], among others [27].

Due to the interpretable power that rules promote, their use can be observed in works that extract them from conventional machine learning models to provide

transparency about the learning process. In the survey [9], the authors list different approaches to extract rules from a trained model, and in [75], the authors present a method to extract logical programs from an artificial neural network. However, these approaches were hidden due to the success of the conventional machine learning algorithms. Still, they are getting evidence since the need to understand the decisions made by a model is increasing. The ability to extract rules from a *DNN* to reasoning and learning is still an open issue in the field.

Overall, inductive rule learning and inductive logic programming represent a powerful approach in machine learning, combining the strengths of inductive reasoning with practical applications across sectors. Its focus on generating understandable rules from data not only aids in effective decision-making but also fosters trust in automated systems by making the reasoning behind predictions explicit. As the field continues to evolve, the integration of rule learning with other machine learning techniques promises to enhance its capabilities and broaden its applicability even further.

1.5 Hypotheses and Research Questions

The ability to identify concepts in a specific layer/neuron in a *CNN* is known as disentanglement [78]. Based on this notion, the initial phase of our research is devoted to identifying semantic concepts learned by a deep-trained model in computer vision, not only at a global level but also about one specific instance in a classification problem. Consequently, the first hypothesis can be formulated as follows:

Hypothesis 1 (H1) *Leveraging disentangled representations can help uncover semantic concepts learned by a trained Convolutional Neural Network globally and for a specific instance.*

To guide the evaluation and assessment of H1, we formulate two Research Questions:

- **RQ1**: How can we extract semantic concepts learned by a *CNN* and measure their relevance in the entire task learned in the deep representation (globally) as well as for a specific input image (locally)?
- **RQ2**: Once we have identified global and local concepts, how can their importance and relevance be assessed for a specific task?

Using the outcome of H1, the next phase in this research is to discover how concepts related to each other, and whether these relations were learned and are latent in the deep representation. Thus, the second hypothesis is:

Hypothesis 2 (H2) *Given a set of concepts, we can use a knowledge graph to identify and validate semantic relations among them.*

Two research questions were formulated to help assessing H2:

- **RQ3**: What relations among disentangled concepts can be extracted by leveraging a generic external knowledge base?
- **RQ4**: How can we evaluate if the relations between the concepts extracted were effectively learned in the deep representation and are therefore used in the classification task?

Humans explain things using the relations among concepts and justify their decision-making processes following a deduction process; thus, using rules can generate a deeper understanding and a more solid base for constructing explanations for humans, integrating learning and reasoning into the process. Upon that, we formulated the third hypothesis, building upon H1 and H2. The result is used to provide a neuro-symbolic approach, helping to enhance the model. The third hypothesis is as follows:

Hypothesis 3 (H3) *Given a set of concepts and basic relations among them expressed as logic predicates, we can learn logical rules that allow deductive reasoning about such concepts.*

Two research questions were formulated to help assessing H3:

- **RQ5**: How can we extract declarative and expressive deduction rules that can partially represent or approximate the decision process of the trained *CNN* in a more explainable way?
- **RQ6**: How well do the extracted rules approximate the model's behaviour, and how well do they generalise?

In order to answer the research questions raised and consequently corroborate the hypothesis, we developed a framework to enhance explainability of trained *CNN* models in image classification tasks. This framework is meant to present a clear set of attributes, which are relate to the user, and, based on the user's prior experiences, to explain what the model learned, demonstrating to the user its intermediate thinking in making a decision. To achieve such output, we claim that the explanation should rely on common sense concepts instead of any technical feature and should present comprehensible relations between the uncovered concepts in a logical form, interpretable by human and intelligent systems.

The proposed approach is inspired by how the learning process unfolds in humans [95]. In fact, we learn not solely through examples but by synthesising knowledge in terms of concepts and their relations, which are then applied to new examples. In turn, new examples serve as inputs to adjust our generalisation based on inductive reasoning. We can identify three main components to the proposed approach: (i) **concept extraction component**, where we generalise the approach for disentangled representation to the identification of concepts the model is likely to have learned, and which play a role in classification of a given input (local concepts) or for an entire classification task (global concepts); (ii) **relation extraction component**, responsible for discovering the top meaningful relations between pairs of concepts learned by the model; and (iii) **rule extraction component** which applies an inductive rule learning approach from identified relations in order to generalise the model's decision process in a declarative and explainable form. The components are represented in Figure 1.2, where we have the two main inputs: the trained

model (the deep representation to be explained) and external knowledge (including domain-specific and common-sense knowledge in form of a knowledge graph), and the output is a set of concepts and relations about what the model learned through rules humans can understand.

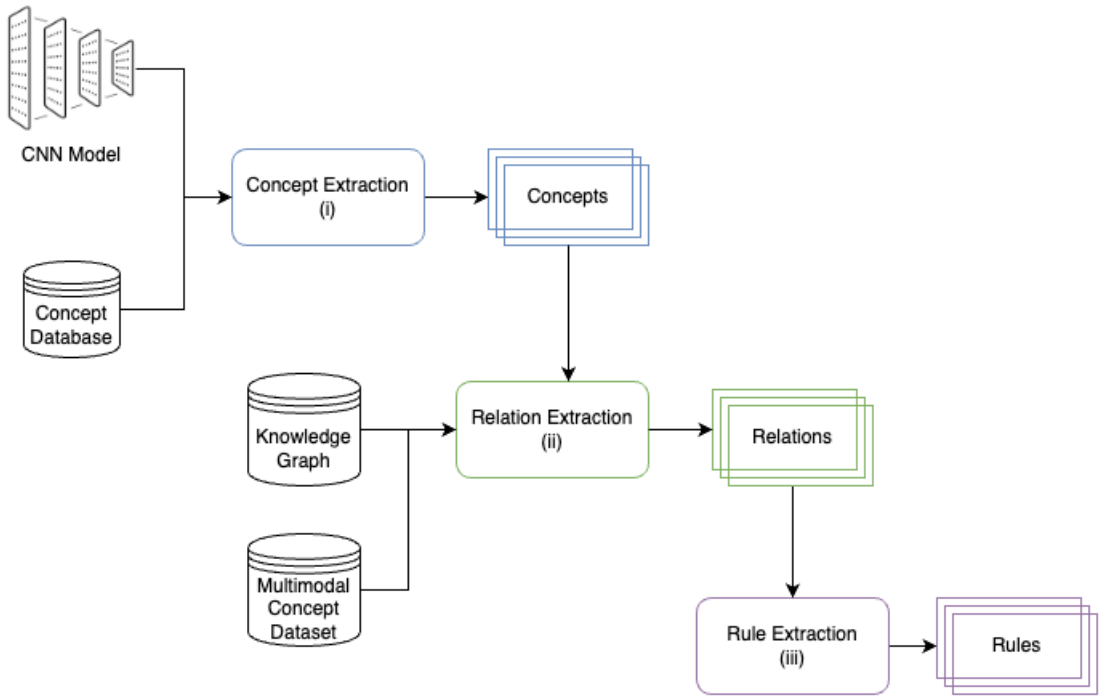

Figure 1.2: Framework for explainability

The proposed framework can be applied to trained image classification models, using different external knowledge to extract the model's inner works. Likewise, other methods can be applied in each component once they follow the component's purpose, highlighting the openness of our framework. This process can also be applied across various AI domains, as it draws inspiration from the human learning process [95], and it addresses the necessity of a pipeline for extracting concepts, relations, and rules to enhance explainability, as discussed in [87]. In this thesis, we presented the framework, providing experiments using a set of inputs and methods, but it does not limit the application scope. We present the work structure in next Section.

1.6 Thesis Structure

In this Chapter we have provided the context, background, motivation for this research. The remainder of this dissertation is organised in the following chapters:

(i) **Literature Review** (Chapter 2): we review state-of-the-art research in Explainable *AI* considering the topics of (i) concept extraction, (ii) relation extraction and, (iii) rule extraction approaches.

(ii) **Concept Extraction from CNNs** (Chapter 3): this Chapter introduces the first component or our framework, which focuses on extracting relevant local and global concepts from a trained *CNN*.

(iii) **Extracting relations from CNNs using Knowledge Graphs** (Chapter 4): building on the previous Chapter, we explore how to extract relations among pairs of concepts identified in the first component of the framework.

(iv) **Neuro-Symbolic Rule Learning for Explainability** (Chapter 5): in this Chapter, we analyse the third and last component of our framework, which aims to use concepts and relations to learn inductive rules that approximate what the model has learned in a declarative representation space.

(v) **Conclusions** (Chapter 6): this final Chapter summarises the research presented in this thesis by revisiting the hypotheses and research questions and to what extent they have been addressed. In this Chapter we also discuss current limitations with the proposed methods and promising directions for next steps and future work.

Chapter 2

Literature Review

This Chapter provides more details on relevant state-of-the-art literature regarding Explainable *AI* (*XAI*) techniques for deep learning models, specifically focusing on explaining Convolutional Neural Networks (*CNNs*). We consider three relevant areas, each of them represented by a section.

We first look at *XAI* for *CNN*, covering different approaches such as visual methods and neuron activation analysis. Since the approaches focused solely on visual methods are not capable of providing a reasonable way to represent an explanation [105], we concentrate on methods based on neuron activation analysis, whereby each neuron can be associated with a common-sense concept, enabling the construction of concept-based explanations. In the second part, we look at the role of a *KG* and its use in explainability [73], reviewing approaches based on explanations of the association between semantic concepts from *KG* and specific neurons. Lastly, we examine the use of logic rules to represent the *DNN's* decision process and the application of such rules for explainability.

2.1 XAI for CNNs

Explainable Artificial Intelligence (*XAI*) has become increasingly important as deep learning models have achieved state-of-the-art performance in various tasks. However, the complexity of these models and their black-box nature make it challenging

to understand their inner workings and decision-making processes. This Chapter aims to provide an overview of the relevant *XAI* methods for this work, focusing on how to explain *CNNs* by using concepts, relations and rules, highlighting their main contributions. According to [5], the *XAI* methods for machine learning models can be categorised into two main groups: intrinsic transparency and post-hoc approaches. Intrinsic involves designing models that are inherently interpretable by their structure, also called shallow models, where the decisions made by the model can be verified following the steps that the model has done. One example can be visualised in the work [33], where the authors used a Decision Trees model to explain what the model learned during the training phase to classify whether a message has been written by a human or a "Bot".

Other models, such as linear regression and K-nearest Neighbours, are examples of simple models for which the explanation is intrinsic [5]. Although these models can provide simple explanations, their performance in downstream tasks may decrease in more complex cases, consequently diminishing the reliability of their explanations. For example, in computer vision tasks, the data type contains complicated characteristics, such as a higher number of dimensions to analyse, which a simple model may struggle to handle in classification tasks. On the other hand, Deep Neural Networks, specifically *CNNs*, have achieved interesting results in this type of task. Since our work focuses on computer vision tasks, the intrinsic models are not the main target.

The *XAI* post-hoc methods aim to explain the predictions of pre-trained models without modifying their architecture or training processes. Within this category, we have two subcategories that separate the post-hoc methods into approaches for agnostic models and specific ones. The approaches that do not consider what kind of model will be used to extract the explanations are called agnostic. We can cite Shapley Additive Explanations (*SHAP*) [81] and Local Interpretable Model-agnostic Explanations (*LIME*) [104] as two model-agnostic methods for image classification as examples.

The *SHAP* approach introduces the *SHAP* values, which offer a unified feature importance measure based on cooperative game theory. They provide a way to understand the contribution of each feature to the prediction by calculating the average marginal contribution of a feature across all possible feature subsets. As for *LIME*, it aims to approximate the black-box model locally with an interpretable model. By perturbing the input data and observing the changes in the predictions, *LIME* constructs a linear model that explains the behaviour of the complex model around the instance being explained. Both approaches explain using feature relevance. Still, the features highlighted lack semantic meaning.

Regarding the post-hoc approaches specifically for *CNNs*, some works have attempted to highlight the model patterns graphically to explain the *CNN* through the visualisation method. Authors in [136] describes a technique for visualising how the model behaves in each layer for a particular image, while [109] uses the gradients of the network output for a single image to improve the pixels that contribute to image classification. Both approaches aid in localising which parts of the image are relevant to a specific class. However, as the image concepts are not declared in the image or the dataset, this visualisation does not represent them, showing a lack of clarity regarding the relations between the concepts (e.g., their importance for the predictions, either individually or in combination) and the low abstraction levels of explanations [106]. Given these issues, the requirement for human experts to interpret what each area means in semantic terms makes visual explanation difficult and challenges the verification of the consistency of highlight features, even across images of the same class in the same dataset.

There are other post-hoc methods applied to *CNNs* to explain what the model has learned. For example, a subcategory called simplification methods uses a surrogate model to clarify the model's inner workings [138]. Another subcategory is the architecture modification methods, which modify the sequence of layers in the model [116] or change the loss function to compel each neuron to learn a specific concept [137]. These show us the variety of methods targeting the same purpose. Neverthe-

less, to achieve interesting results using them, we must simultaneously change the model's structure, which could affect its performance; still, this does not provide a semantic and reasonable explanation of the deep representations learned by the model.

We believe that a more trustworthy and feasible way to understand the deep representations from the trained model is to use feature relevance methods and translate the features into semantic meaning so that human experts can understand what has been learned. Feature relevance methods aim to calculate a relevance score for a model's managed features and clarify its inner workings. These scores quantify a feature's affinity (sensitivity) for the model's output. Comparing the scores of different features unveils the importance the model assigns to each one when producing its output. We focused on this method for this thesis, and in the following Subsection, we present related works.

2.1.1 Feature Relevance Methods

In order to emphasise the relevance of features learned by a model, linking each feature to a meaningful concept, three main techniques address feature relevance in different ways: one is through disentangled representation; another involves training a separate model to learn predefined concepts; and the last makes use of multi-modal approaches to extract concepts.

Feature Attribution based on Disentangled Representations

Disentangled representation is a method that proposes dividing each characteristic (of an image) into carefully specified variables and encoding them as distinct dimensions. The idea is to emulate humans' swift intuitive process[1]. This method can characterise the semantic concepts gained by a model throughout its training phase. We list the first works that employed this strategy to characterise concepts learned in deep representations for computer vision, which are still used as the basis for this

[1] https://deepai.org/machine-learning-glossary-and-terms/disentangled-representation-learning

field of research.

Networks Dissection is a method for extracting meaning from each layer or filter, using the distillation approach to explain a *CNN*. Authors in [141] claim that a *DNN* may spontaneously learn disentangled representations. In order to demonstrate this, they developed a framework for connecting human notions to each filter in a *CNN* model (Figure 2.1). The objective is to provide meaningful labels for individual filters. The initial stage was to generate the *Broden* dataset, which contains pixel-annotated low-level notions such as colours and high-level concepts such as objects. They then used a trained model and passed through the *Broden* dataset to assess each filter and compare the binary map from each image with each filter activation map. If the convolutional filter is strongly activated in parts of the picture containing a human-labelled notion, the authors claim that the filter is "searching for" that idea or concept. This approach is still being used for different scenarios; for example, generating new images using a generative adversarial network [15], where knowing which neuron represents a specific concept is important for modifying an existing image.

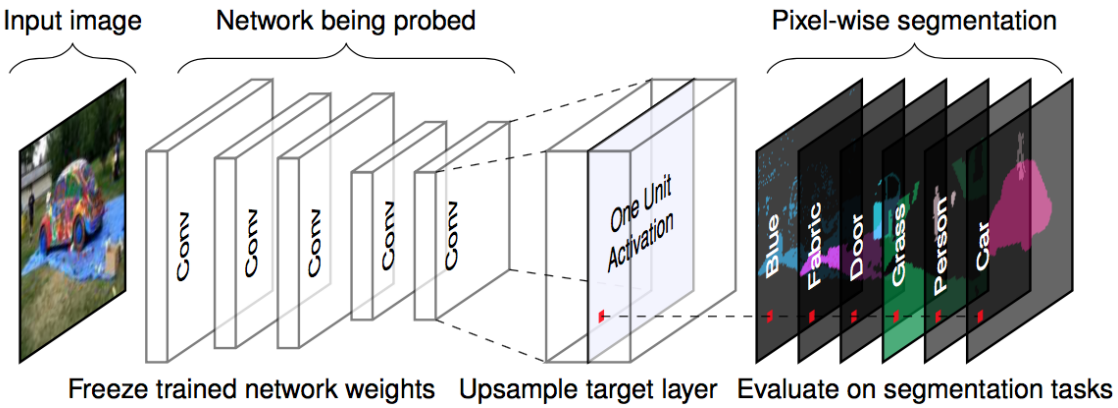

Figure 2.1: Network Dissection framework ([141])

Examining different *CNN* designs, authors discovered certain important notions, such as the number of unique concepts for each layer in each architecture and the increasing number of objects in deeper convolutional layers. In [138], the authors suggested learning a decision tree from a trained *CNN*, detailing the exact reasons for categorisation at a semantic level. The proposed technique describes which image

components activate, which filters are used for categorisation, and how much each part contributes. The authors use the simplification method in this approach to extract a straightforward explanation from a complex model. The first part of the approach is training a *CNN* with disentangled filters on the high convolution layers. Each filter learns a specific concept and associates each one with semantic meaning, as there are no annotations for the concepts.

This approach is presented in [137], where a loss function is applied to each filter in the top convolutional layer. The trained disentangled filters extract information from each image and input it into a decision tree that understands its composition. There is no link between a filter and a human notion at this time. Thus, the authors use the *CUB-200* [103] dataset to assign a concept to a specific filter. They concentrate on a single topic (bird) and only use concepts relating to that topic. This method differs from [141] because it does not employ an extensive concept dataset to assign the concept to each filter. However, it can be used to search for concepts unavailable in the Broaden dataset. An extension of this work is presented in [22], where, instead of using a decision tree, the authors build two different networks - teacher and student - in which the student network is meant to learn the teacher's behaviour. Although the approach does not rely on an extensive concept database, the explanation is generated from a surrogate model that tries to maximise the similarity between the two models. Still, as complex as the main model becomes, with deeper layers, for example, shallower models may fail.

Feature Attribution based on Separately Learned Concepts

When we look for the other group that aims to extract feature attribution based on separately learned concepts, a relevant work in [61] proposes determining how human notions influence categorisation results. The authors defined and developed the *CAV* (Concept Activation Vectors) to transform a neural network's internal state into human-friendly notions - a common-sense concept. The method is useful because a human concept, such as "stripes", may be shown to impact the "zebra" class

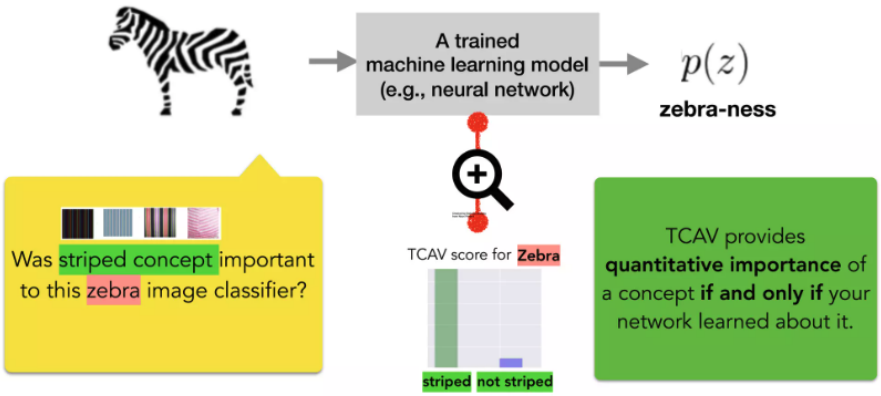

Figure 2.2: Importance of the concept "stripes" for the class "zebra" [61]

(Figure 2.2). The core idea is retrieved from a trained model, a vector characterising a particular concept is created, and then a directional derivative is used to assess concept sensitivity for a specific class. This method provides a local explanation for a particular concept within a class, which is particularly suited to users who already understand which concepts apply to a given class and want to identify among a set of such concepts which ones are more descriptive of that class from the perspective of the deep representation. This validates which concepts, among a set of pre-defined one learned via a separate linear model, most affect a classification.

The *CAV* paper was extended on the *SPACE* algorithm [96], which focuses on concept extraction for scale-sensitive models and data, particularly in contexts such as automatic quality control. It provides a new method for extracting and testing concepts, aiming to improve the understanding of trained models and enhance defect detection capabilities. It was also extended in other works, such as in [133], to specify which concept was learned, the causal effect of a concept in relation to a class [134], and for model debugging [2]. Similarly to the *CAV* work, the explanation generated by these approaches still relies on the necessity of knowing which concept should be analysed, focusing only on a local example.

Multi-modal approaches for Concept Extraction

The last method presented takes advantage of multi-modal approaches, such as a combination of images and text in a pre-trained model, to extract meaningful

concepts from a *DNN*. Recently, models integrating visual and textual data have been developed to create a multi-modal understanding of concepts. An example is Contrastive Language-Image Pre-training (*CLIP*) [98], a model that operates by jointly training an image encoder and a text encoder on a vast dataset of 400 million (image, text) pairs sourced from the internet. The core of *CLIP's* functionality lies in its ability to learn a shared embedding space where images and their corresponding textual descriptions can be compared (Figure 2.3). During training, the model maximises the cosine similarity between correct image-text pairs while minimising it for incorrect pairs, facilitating a robust understanding of how visual and textual information relate. The architecture of *CLIP* employs a dual-encoder system, where both the image and text are processed separately before being projected into a shared latent space. This design enables the model to effectively leverage natural language supervision, making it capable of generalising across various tasks such as image classification, semantic search, and image generation. The model's efficiency is notable; it can achieve competitive performance on numerous benchmarks without manually labelled datasets, which are often costly and time-consuming to create. However, the text representing the concepts must be provided manually, which is a barrier to a more extensive acceptance.

In [93], the authors introduce *CLIP-Dissect*, a technique that describes the hidden neurons of a deep neural network, using the *CLIP* model to automatically link concepts to hidden units without the need for a labelled dataset of pre-defined concepts. Although this work has achieved interesting results in associating concepts with neurons, it still does not capture relations between concepts. Consequently, concepts are considered independent, which limits the ability to interpret the model's behaviour. This method also relies on a predefined list of concepts, which varies from one domain to another.

Another method to gather feature importance from a trained model is Concept Bottleneck Models (*CBMs*). In [64], the authors introduce an approach incorporating a bottleneck layer before the fully connected layer, forcing the network to

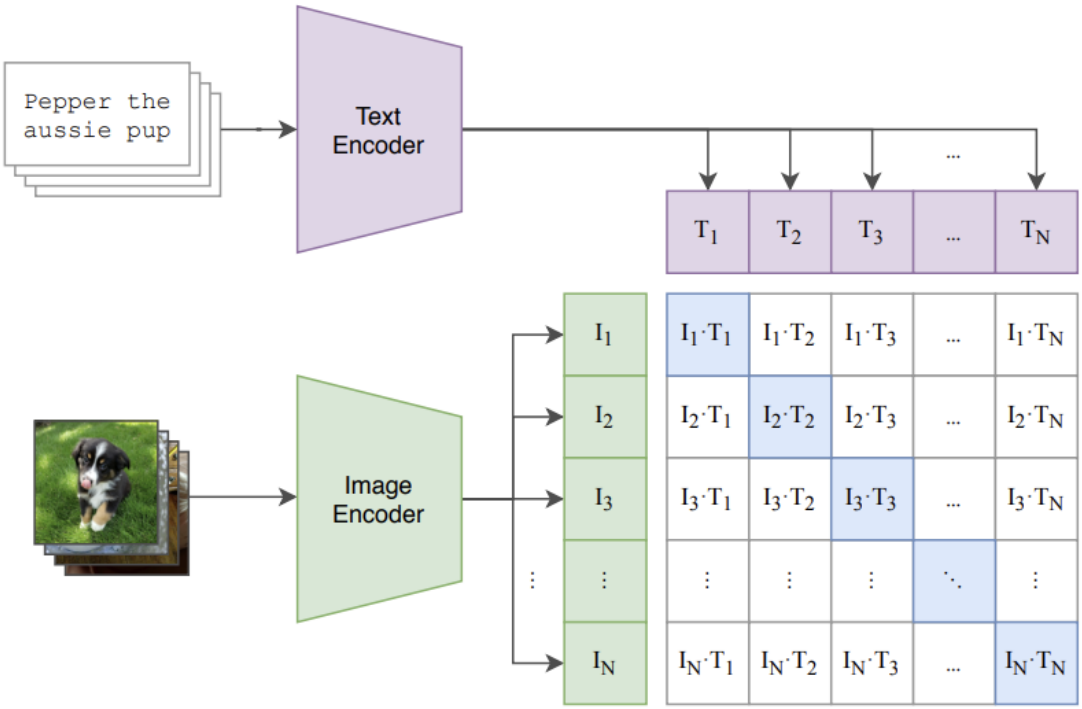

Figure 2.3: CLIP Embedding Process ([98])

compress information into a compact set of interpretable concepts. Results demonstrate that *CBMs* achieve competitive performance on standard benchmarks while providing interpretable representations. However, limitations include the potential loss of fine-grained details during concept compression, as some concepts may be left out, and challenges in determining the optimal number and nature of concepts to represent complex data effectively.

In order to address the use of *CBM* without relying on a pre-defined number of concepts, authors in [94] introduce a variant of *CBMs* called *Label-free CBM*. The approach was designed to work in a label-free setting where direct access to ground truth concepts may be limited or unavailable. In *Label-free CBM*, concepts are learned directly from raw class labels using a large language model [1]. Then, they are linked to hidden units based on the concept representation similarity between the *Clip* model encode and the neuron activation map. Results demonstrate that *Label-free CBMs* achieve competitive performance on various tasks without relying on labelled concepts. Although this approach overcomes one limitation from

[93] (the need for predefined concepts), it still faces the potential loss of concept representation due to the bottleneck process and the lack of a reasoning explanation about the relation of the concepts.

In this Section, we present relevant literature on *XAI* approaches designed to enhance the transparency of *CNN* models by providing human-understandable interpretations through feature importance. Although the various methods achieve some level of concept extraction, the drawbacks — such as a strong reliance on a specific dataset for the concepts, the manual setting of concepts, and the loss of representation — contribute to the lack of reasoning behind explanations. Aligning with these issues, focusing solely on analysing local explanations, regardless of how they generalise, can be regarded as a significant concern. Nevertheless, it is necessary to explore this area further, specifically because it is not immediately clear how concepts are related. As humans, we accept and provide explanations based on relations between concepts rather than the pure relevance of concepts to a given decision. Therefore, the ability to characterise such relations has great potential to increase acceptance and trust in these models.

2.2 Knowledge Graph and Explainability of Deep Neural Networks

The integration of knowledge graphs (*KGs*) into *XAI* frameworks for *DNNs* is increasingly recognised as a pivotal advancement in enhancing model interpretability [73]. *KGs* serve as structured representations that encapsulate relations between concepts, enabling a more nuanced understanding of how *DNNs* arrive at their predictions. By providing a semantic layer that connects abstract neural representations to real-world knowledge, *KGs* facilitate the extraction of meaningful insights from complex models. This is particularly important in high-stakes domains where transparency and accountability are paramount, such as healthcare and finance.

KG are inherently suitable for describing relations between concepts due to their

ability to represent entities and the connections between them in a structured format. They can encode various types of relations, including hierarchical, associative, and causal links, which are essential for reasoning about the interactions within a dataset [108]. Recent research has demonstrated that *KGs* can effectively enrich *DNNs* with external knowledge, thereby improving the quality of explanations generated by these models [101]. For instance, co-activation graphs can reveal the relations between neurons in *DNNs*, linking their activation patterns to specific concepts represented in a knowledge graph [53]. This approach not only aids in understanding the model's decision-making process but also enhances the interpretability of its outputs by grounding them in familiar semantic structures.

Furthermore, there is significant potential in combining knowledge graphs with neuron activation analysis to augment data with contextual knowledge. Researchers can uncover deeper relations and dependencies that inform model behaviour by analysing neuron activations alongside *KG*-derived insights, thus providing users with a comprehensive understanding of model predictions. In most of the cases, the *KG* is created using user experts, which create graphs align to their knowledge. However, there is a *KG* called common-sense knowledge graph which can be integrated into this framework to further enhance interpretability and applicability across diverse domains.

Overall, leveraging *KGs* in conjunction with deep learning addresses the challenges of explainability and paves the way for more robust and trustworthy *AI* systems. In the context of this thesis, we aim to use *KGs* to extract meaningful relations and concepts from a trained model, so they can use to generate explanations. From the perspective of the use of *KGs* for *XAI*, this falls under post-model explainability according to a recent survey [100]. According to the survey, the majority of approaches look at pre-model or in-model explainability, while post-model explainability methods focus on using *KGs* in conjunction with trained model to combine learning and reasoning as opposed to generating understanding of the model's representation for explanation generation. For this reason and due to the scarcity of work

in this area, in this section we focus on discussing what common-sense knowledge graphs can be leveraged in this research in order to contribute to the state-of-the-art on post-model explainability via Knowledge Graphs.

2.2.1 Common-sense Knowledge

Common-sense knowledge refers to information that people consider apparent or intuitive, such as the concept that fire is hot or that running on sand is more complex than running on a hard surface. Collecting and expressing this information is critical for artificial intelligence applications because it helps machines understand and reason about the world as humans do.

The *ConceptNet* [115] is an example of a Common-Sense Knowledge Graph. It is part of a larger initiative to create large-scale knowledge graphs (*KGs*) that may be used to improve natural language processing and other artificial intelligence applications. *ConceptNet 5*[2] improves upon previous work in this field, and it possesses several characteristics that make it a valuable resource for professionals and academics. *ConceptNet* is designed to capture relational knowledge in diverse scopes, making it a valuable resource for various *AI* applications.

ConceptNet combined elements from works such as *WordNet*[3], *OpenCyc*[4], *DBpedia*[5], among others, to provide a wide range of linkages between concepts in many domains. It also supports multilingual knowledge representation, allowing it to represent knowledge in many languages.

The Common-sense Knowledge Graph (*CSKG*) [57] is another relevant work that used seven very diverse and disjoint sources: a common-sense knowledge graph *ConceptNet*, a general-purpose taxonomy *Wikidata*[6], an image description dataset *Visual Genome*[7], a procedural knowledge source *ATOMIC* [107], and three lexical

[2]https://conceptnet.io/
[3]https://wordnet.princeton.edu/
[4]https://www.qrg.northwestern.edu/Resources/resources_index.html
[5]https://www.dbpedia.org/
[6]https://www.wikidata.org/wiki/Wikidata:Main_Page
[7]http://visualgenome.org/

sources: *WordNet*, *Roget* [63], and *FrameNet*[8]. *CSKG* offers certain advantages to *ConceptNet*, such as more nodes and edges and being more organised and consistent. Nonetheless, the critical distinction is that *CSKG* is limited to common sense knowledge, whereas *ConceptNet* may be employed in various contexts.

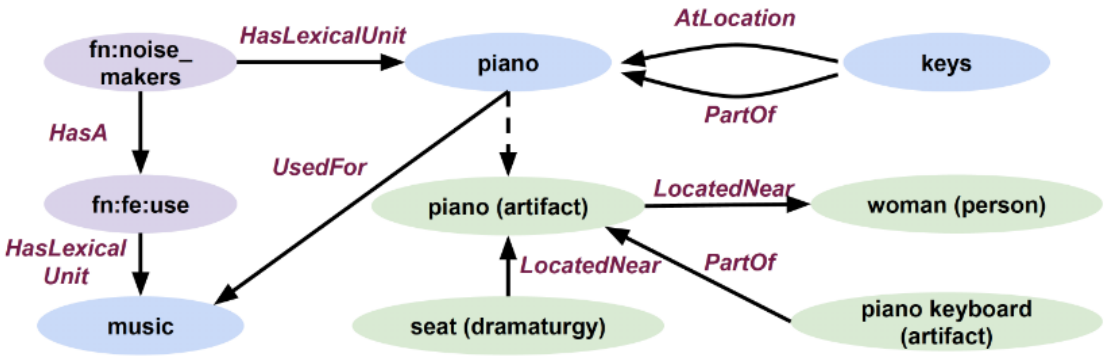

Figure 2.4: CSKG example ([57])

Some concepts and relations are illustrated on *CSKG* in 2.4. The object "keys" has a relation "Part Of" with the object "piano", which has a relation "Used For" with the object "music". The knowledge that a piano has keys and is used for music is knowledge that a person can comprehend with minimal prior knowledge. It might be a plausible method to describe how a model learns a given task.

KG has also been employed to develop a simple method for understanding the relation between objects in a picture. The *ConceptNet* was used in the images following the classification work in [56] to emphasise the local relations, providing an enrichment element to the model. Nonetheless, this procedure is manually curated to align the concepts [119], and it does not aim to explain how the model learned those concepts and relations.

The authors' recommended technique in [82] comprises expressing objects and their connections using a *KG*, which provides a structured and formalised representation of the objects and their properties. However, the *KG's* alignment with the concepts and the manner in which the model comprehends their relations remains unresolved.

[8]https://framenet.icsi.berkeley.edu/fndrupal/

2.3 Inductive Logic Programming

Creating a human-friendly explanation and providing a language for expressing structured knowledge, as well as communicating high-level concepts [54], is a challenge in explanation generation. One approach that can be employed to overcome the language structure problem is to use rules. Rules can be simple *IF-THEN* statements [49] or a logical program, such as Inductive Logic Programming (*ILP*). Creating rules using human-interpretable concepts extracted from a *DNN* has significant value for human experts who are not machine-learning specialists. Rules can assist developers in debugging a model at a high level to identify areas where the model can be improved, while human experts can provide direct feedback on the validity of the deduction processes made explicit by the rules.

Rule extraction from *DNNs* has emerged as an important area of research in recent years, driven by the need for interpretability and explainability in deep learning models. Rule extraction methods for *DNNs* can be broadly categorised into three main approaches: decompositional, pedagogical, and eclectic [48]. Decompositional methods target the internals of the network by extracting rules from each hidden and output unit, then aggregating them into global rules that link inputs to outputs. Pedagogical methods focus on the function learned by the network and treat it as a black box. The extracted rules describe the behaviour of the network's output in terms of inputs without utilising the internals of the network. Eclectic methods are hybrid, examining the units of the network while extracting rules that directly describe the relation between the inputs and the outputs [86]. Following this, we present the most relevant works in each category for this thesis.

Decompositional Approaches

DeepRED [143] is an algorithm designed to extract rules from deep neural networks. It works by decomposing the network into adjacent layers and then using a decision tree induction algorithm to extract rules from pairs of layers. The main advantage of *DeepRED* is its ability to handle deep neural networks, which many earlier rule

extraction methods struggled with. However, *DeepRED* can be computationally expensive and time-consuming, especially for larger networks. It may also produce complex rule sets that are difficult to interpret for very deep networks. *REM-D* [112], another decompositional approach, is part of the *REM* (Rule Extraction Methodology) framework, focusing on extracting rules from deep neural networks. It breaks down the network into adjacent layers and extracts rules from each pair. *REM-D* benefits from the noise removal property of neural networks and considers both hidden and input features in its rule extraction process. The main advantages of *REM-D* include its ability to produce accurate and comprehensible rule sets while being relatively efficient in terms of time and memory usage. However, like other decompositional methods, it may face scalability issues with very deep or complex networks.

ECLAIRE [135] (Efficient CLAuse-wIse Rule Extraction) is designed to extract rules from deep neural networks in a more scalable way than methods such as *REM-D* or *DeepRED*. It aims to produce better-performing and smaller rule sets, making it particularly suitable for large models or training sets. The main advantage of *ECLAIRE* is its efficiency and scalability, allowing it to manage complex models that might be intractable for other methods. However, as with any rule extraction method, there may still be a trade-off between fidelity to the original model and the simplicity of the extracted rules. Another approach is *EDICT* (Extracting Deep Interpretable Concepts using Trees) [127], which focuses on extracting interpretable concepts from deep neural networks using decision trees. It aims to provide a more intuitive understanding of the network's decision-making process by identifying key concepts learned by the model. The main advantage of *EDICT* is its ability to offer high-level, conceptual explanations of a neural network's behaviour. However, it may not capture all the nuances of the network's decision-making process, especially for very complex models.

Additionally, the work of [20] is an example of how a *Deep Belief Network* may derive rules from an image classification problem. Even though it is a strategy that

develops global rules; that is, a rule that describes an entire class, it is not composed of human concepts; instead, it employs a feature and a threshold linked to it. This style of explanation is appropriate for a machine learning professional but not for the end user. Overall, decompositional methods may perform better when extracting rules from a shallow or non-complex model, but their capability can be compromised in a computer vision task. As these approaches rely on algorithms, such as decision trees, their ability to provide a more complex rule is diminished.

Pedagogical Approaches

TREPAN [26] is a pedagogical rule extraction method that treats the neural network as a black box. It uses queries to the trained network to build a decision tree that approximates the network's decision boundaries. *TREPAN* can handle continuous and discrete attributes and uses an oracle to answer queries about the network's behaviour. The main advantage of *TREPAN* is its ability to work with any neural network architecture without needing access to its internal structure. However, as a pedagogical method, it may miss important internal structural information of the network. Another pedagogical approach is *RxREN* [8], a rule extraction method that combines rule extraction (Rx) with relevance (RE) and novelty (N) measures. It aims to produce more compact and relevant rule sets by identifying and pruning less important inputs or rules. The main advantage of *RxREN* is its ability to generate more concise rule sets while maintaining good predictive performance. However, the pruning process may potentially result in some loss of information from the original network.

Pedagogical approaches claim that they are agnostic methods, as they do not take into account the *DNN's* inner workings, focusing on the input and output while approximating the learned process. They potentially miss important model information, which can be detrimental when the input comprises complex data, such as images.

Eclectic Approaches

The authors in [18] proposed extracting rules from a *CNN* using a Discretised Interpretable Multi-Layer Perceptron (*DIMLP*), a layer in which the staircase activation function serves as the activation function for each neuron. The core idea behind *DIMLP* rule extraction is the exact localisation of axis-parallel discriminative hyperplanes, indicating that the input space is partitioned into hyper-rectangles encoding propositional rules. This approach generates rules for each route by feeding them into a decision tree. The investigation in this work was carried out using textual data, and a set of rules was created based on how the n-grams contributed to a specific class. The authors suggested generating propositional rules from a *CNN* by approximating the model with two subnetworks from which rules may be developed in [19]. This work is an expansion of [18], now applied to an image classification problem, and it illustrates the rules derived from a *CNN* while highlighting the areas in a collection of images corresponding to each rule. Nonetheless, the rules are connected to the image pixels, which do not reflect a human notion, making it challenging to develop an explanation generalising the idea.

In order to extract complex rules where a reasonable process can be conducted on [123], neuro-symbolic approaches have been combined with neural networks to achieve this objective. *LNNs* [110], which are a neuro-symbolic approach that combines neural networks with logical reasoning, provide an example. They maintain a strong connection to classical Boolean logic while allowing for gradient-based optimisation. *LNNs* can be used to induce first-order logic rules from neural networks. The main advantage of *LNNs* is their ability to bridge the gap between neural networks and symbolic *AI*, potentially offering more powerful and flexible rule extraction capabilities. However, they may require specialised training procedures and may not be directly applicable to pre-trained standard neural networks.

NeurASP [130] is an extension of answer set programs that integrates neural networks with Answer Set Programming (*ASP*). It treats neural network outputs as probability distributions over atomic facts in *ASP*, allowing for the combination

of sub-symbolic and symbolic computation. The main advantage of *NeurASP* is its ability to incorporate complex semantic constraints and apply symbolic reasoning to improve neural network perception. However, it requires expertise in both neural networks and *ASP*, which may limit its accessibility.

In conclusion, the research presented in this chapter emphasises that the requirement for a high-level method to represent what the model has learned remains an open issue in ensuring transparency throughout the learning process, particularly in the field of computer vision. This thesis aims to provide a novel framework in which each component addresses the specific limitations highlighted in this chapter. The first component focuses on the extraction of high-level concepts by ranking the most significant ones for the classification task, thereby overcoming reliance on low-level features for understanding what the model has learned. The second component focuses on extracting high-level relationships identified by the model, a method not yet applied to understanding deep representations from a *CNN* model, offering a transparent means of understanding the knowledge encoded during the process. The last component presents a method for making the learning process more explicit, by providing a high-level structure through which humans and intelligent systems can interact, thus paving the way for the effective integration of neural learning and deductive reasoning.

Chapter 3

Extracting Concepts

3.1 Introduction

This Chapter presents the first component of the framework for understanding a trained *CNN* model's deep representation. Through this step, we define an approach to extract human-understandable concepts from a trained model, representing aspects the model understands for each image (local concepts) and each class (global concepts).

Most approaches for interpreting directly the output of a trained *CNN* in a classification task have focused on visual cues and, more specifically, attention-based methods. For example, works in [136, 109] have highlighted image pixels or areas contributing to a specific classification. The authors in [136] describe a technique for visualising how the model behaves in each layer for a particular image. Both approaches aid in localising which parts of the image are relevant to a specific class.

However, these methods do not provide any indication of how the highlighted set of pixels aligns with the corresponding concept illustrated by those pixels [128]. This lack of explicit concept-pixel association results in attention maps that can be ambiguous when looked at by humans to justify a prediction. From the human perspective, these attention maps may draw the attention of the human evaluator onto regions of the image that are not conceptually relevant, which does not help in understanding [60]. This is particularly problematic when we want to understand

the reason why a model makes a mistake that cannot immediately be interpreted by looking at the image only. One example is illustrated in Figure 3.1, where the attention maps highlight the parts of the image that determined the model's outcome for two different classes (one is correct and another is incorrect). This image is taken from [105], and we want to report it here to illustrate how challenging it could be for a human to interpret the reasons why the third image has been classified as a *Transverse Flute* and what does the highlighted portion of the image represent.

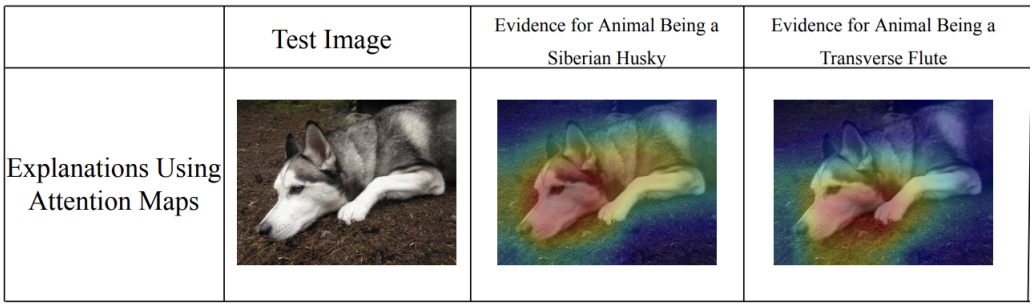

Figure 3.1: Explanation example based on attention maps ([105])

In order to tackle the limitations of visual approaches, other explainability methods have been proposed in recent years. The authors in [88] present other strategies, such as textual justification, simplification, and feature relevance. When we learn textual data along with visual data, the use of textual justification enhances understanding of the model's results; this has been used in the medical domain to clarify categorisation by combining image and textual diagnostics. The simplification strategy involves creating a white-box model from a complex model to achieve performance while simplifying the explanation. Lastly, feature relevance is a method that considers each feature's contribution to the model's output to describe the learning process. Additionally, the authors in [52] present a strategy that employs human expertise to explain how the model has learned in a manner that a layperson may comprehend, rooted in real-world principles.

As explained in Section 2.1.1, disentangled representation is a method that proposes dividing each characteristic in an image into a semantic representation, thereby allowing low-level features to be interpreted as real-world concepts. Using a feature-relevance strategy, where each feature has a semantic representation through the

disentangled method, we can convert low-level features into real-world concepts, thus representing which concepts the model has learned. By ranking them, we intend to present the most relevant concepts from a single image - also referred to as an instance - as well as from an entire class, which is a collection of instances belonging to the same category or group (therefore having the same label). We also rely on the premise that each neuron or filter in a *CNN* corresponds to one concept, as presented in [141].

This Chapter will present our approach based on the idea that if we can successfully disentangle representations by associating filters to concepts, we can then use a linear classifier to rank those concepts so that we can identify what characterises an instance and, more generally, a class from local disentangled representations. We aim to provide a list of the most relevant semantic concepts that have affected the model's outcome for a given instance, as well as for a specific class. This is the first step in our framework, where we aim to provide a human-understandable and self-explanatory representation of a trained *CNN* model in terms of the semantic concepts with which this model associates its instances and/or classes. We plan to build upon the ability to identify semantic concepts as disentangled representations and to design a ranking based on their semantic relevance to an instance or a class. By leveraging such a ranking, we can further extract semantic relations (Chapter 4) and learn logical rules from deep representations (Chapter 5).

We can summarise the key contributions of this Chapter which focuses on the first component of our framework as follows: (i) a method to extract the semantic concepts from a trained *CNN* model in a form that enables humans to understand what the model has learned (Section 3.2) **(RQ1)**; (ii) a ranking algorithm to compute the importance of the concepts extracted in relation to a specific instance as well as an entire class (Section 3.3) **(RQ2)**. Figure 3.2 depicts the elements of the concept extraction component developed in this Chapter. The component uses two datasets: one containing the high-level concepts labelled (the *Broden* dataset) and the other selected from different domains related to the classification task and used

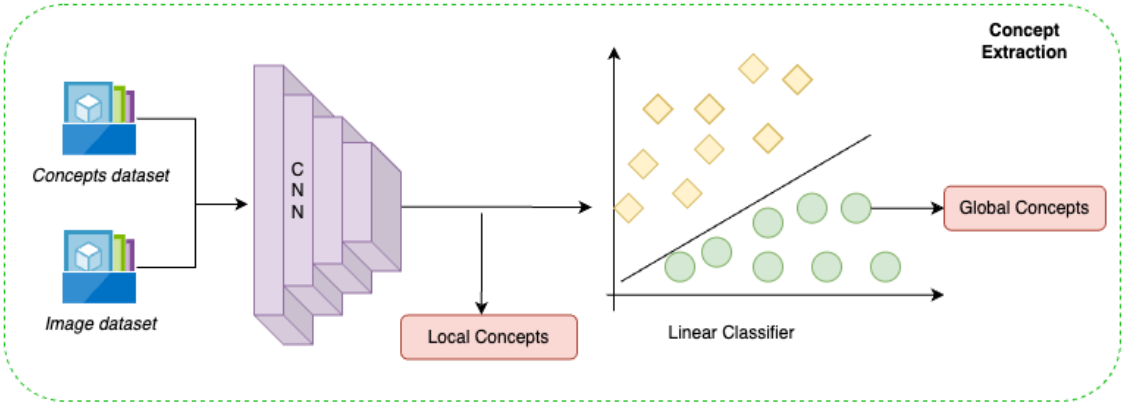

Figure 3.2: Concept Extraction Component

for probing these concepts (*Action40* and *CIFAR-10*). Based on these datasets, a trained *CNN* model is used to link the concepts from the first dataset to each neuron/filter. The second dataset is subsequently used to extract the concepts associated with each instance. Finally, a linear classifier is introduced, which gathers the local concepts and generalises them to perform the classification task.

The novel part of this component is the use of a linear layer (instead of a fully connected layer) subsequent to the local concept extraction. This linear classification has the function to derive global concepts and rank them in order to assess their importance in the classification task. The formula used to calculate the significance of the concept is also an innovative aspect of the process. The source code for this component is available online[1].

3.2 Concept Extraction (RQ1)

The first step in this component is to extract semantic concepts about a class or a single instance from a trained *CNN* model. To do that, we used previous research [141] that has already obtained promising results on semantic concept identification for a trained *CNN* model as a starting point. One of the outcomes of such work is the ability to quantify how interpretable a *CNN* is by discovering how individual hidden units align with semantic concepts at each hidden layer. Concepts

[1]https://github.com/EricFerreiraS/disentangled_representation-concept_ranking

were identified as part of six categories: object, part, material, colour, texture, and scene. The architecture that identified more unique concepts among those tested were *ResNet-152* [51] (Appendix B.2.1) and *DenseNet-161* [55] (Appendix B.2.3), as indicated in Figure 3.3; therefore, these are the architectures we adopt. In order to verify whether a smaller and, therefore, less computationally intensive *ResNet* model (with fewer residual layers) would yield comparable results, we also included *ResNet-50* (Appendix B.2.2) in our comparison. We extended the *Network Dissection* [141] technique for concept detection by identifying and connecting such concepts to output classes and individual input images (or instances). We focused on the last *CNN* layer to maximise the number of unique high-level semantic concepts discovered; as more different concepts are harnessed, more concepts may be connected with local or individual examples.

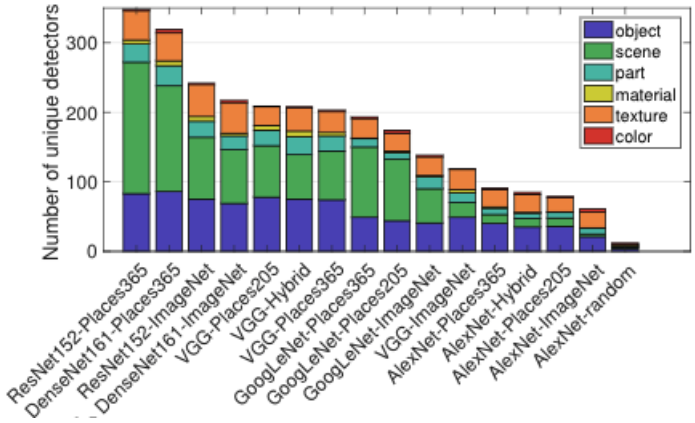

Figure 3.3: Unique Detectors for each *CNN* Architecture ([141])

We start from the semantic concepts identified by *Network Dissection* [141] (presented in 2.1.1) from the trained *CNN* model and build upon the relation between the convolutional filter and the semantic concept from the *Broden* [14] dataset. A transfer learning approach enables the adaptation of the *Network Dissection* method to different datasets (and classification tasks) and allows us to determine which filters are activated by each new input. We retained the weights learned during the training process, replacing only the fully connected layer to preserve all the knowledge acquired by the model. With this approach, the concepts learned from *Network Dissection* are associated with new input images: the top K highest-scoring filter-

s/neurons for each input image are selected as the identified concepts, considering the mean of each activation map.

Note that an activation map is a matrix that represents which image part was activated after the convolution function. It can be represented by a matrix $A_{M\times N}$ of the elements a_{ij},where $M = 1..i$ and $N = 1..j$. We define the mean of the activation map matrix $M_{Activation\_map}$ as follows:

$$M_{Activation\_map} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} a_{ij}}{\#E} \tag{3.1}$$

where $\#E$ is the number of elements of A.

We then rank the K filters/neurons from highest to lowest based on the mean activation map for each image, assuming that the highest value identifies the most representative concept contained in an image. As a result, the approach produces a list of K different semantic concepts that are considered meaningful for each image. A linear classifier, using the model-extracted features as input for the global concepts, is applied to the same dataset for each class; subsequently, based on feature significance, we gather which semantic concepts better separate classes and can therefore be identified as the most relevant concepts at a global level. The datasets used for the investigation are the *Action40* (Appendix B.1.2) dataset [131], which contains action images labelled for 40 different action classes, and the *CIFAR-10* (Appendix B.1.5) dataset [66], containing tiny images for 10 different classes.

3.3 Concept Importance Evaluation (RQ2)

To evaluate the importance of the concept, we assess the precision between the ranked semantic concepts from local instances and the ranked semantic concepts for their class using the list of top K high-scored concepts from local and global examples. We compare the concepts for each local example (image) belonging to a specific class to the top concepts that best linearly separate the class. We consider

the globally rated concepts to make sense of the local ones if at least one concept is common between them. The formula for this can be expressed as follows:

$$P_c = \frac{\sum C_{l_c,g_c}}{\#L_c} \tag{3.2}$$

where P_c is the precision of the specific class c, $\sum C_{l_c,g_c}$ is the sum of the instances where the global and local sets shared at least one ranked semantic concept for class c, and the $\#L_c$ is the number of local instances that belongs to class c. This metric was derived from the standard precision formula.

This metric indicates how well the global characteristics, separated linearly, reflect the semantic concepts acquired by the model for each class. This is an intuitive metric that gives us a good indication of what the model has learned about a class concerning its instances in terms of semantic concepts that human experts can understand. Following this approach, the next section will present our experimental evaluation of the method for concept extraction.

3.4 Experimental Evaluation

3.4.1 Setup

As mentioned in Section 3.2, we used the *Network Dissection* technique to extract local concepts and applied a linear model (SVM[2]) to extract the global concepts per class. We ran the algorithm using a 5-fold cross-validation, using the learning rate (C) equal to 0.001[3]. The task we consider in our investigation is action recognition based on the *Action40* dataset [131] (Appendix B.1.2), which includes 9,532 images across 40 action classes, (4000 used for training and 5,532 used for testing) and the well-known *CIFAR-10* [67] (Appendix B.1.5) dataset, containing 60,000 (50,000 used for training and 10,000 used for testing) small images across ten classes. The

[2]implementation from https://scikit-learn.org/dev/modules/generated/sklearn.svm.LinearSVC.html

[3]grid-search technique from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

concepts associated with each filter in the *CNN* model are provided by *Network Dissection*, which was trained on the *Imagenet* dataset [31], considering only a limited set of categories, namely *object*, *part*, *material* and *colour*. We collected only the concepts identified in the *CNN's* last layer, and the unique number of concepts identified from the trained model in the *Imagenet* dataset is presented in Table 3.1. Then, using a transfer-learning approach, freezing all the trained layers and replacing the fully connected layers with the linear classifier, we used the *Action40* and *CIFAR-10* datasets to capture the concepts learned for this data based on the *Network Dissection* results. Table 3.1 also presents the number of unique concepts detected after the transfer-learning approach, which extends from the concepts learned by the trained model (*Imagenet*). The closer to the number of unique concepts from the trained model, the closer the target dataset can be represented by the trained model.

	# Unique Concepts Imagenet	# Unique Concepts Action40	# Unique Concepts CIFAR-10
ResNet-152	162	152	153
ResNet-50	161	151	149
DenseNet-161	186	128	127

Table 3.1: Unique concepts identified per model and dataset, using the *Network Dissection* approach.

The local semantic features from the new data were recovered using the mean of the activation map from each filter in the final layer, as per Formula 3.1. The same formula was used for global concepts, i.e., concepts for each class, but this time, it was applied to feature extraction for classification input. Following the intuition in [61] that meaningful higher-level concepts may be more straightforward to grasp, we used the *SVM* and selected the model with the best F1 score. The classification algorithm produces a confusion matrix, as presented in Figures 3.4 and 3.5, which displays the precision obtained for each class by the datasets for the

ResNet-152 model's output. We focused on the *ResNet-152* because it identified more concepts between the datasets in relation to the trained one (Table 3.1) and achieved relevant performance during the classification task (Table 3.2). The linear model metrics for each architecture and dataset in the class separation (classification task) are presented in Table 3.2.

		Accuracy	Precision	Recall	F1
ResNet-152	Action 40	81.46%	80.05%	80.73%	79.95%
	CIFAR-10	**92.24%**	**92.23%**	**92.25%**	**92.23%**
ResNet-50	Action 40	77.97%	76.85%	77.36%	76.97%
	CIFAR-10	90.45%	90.43%	90.47%	90.44%
DenseNet-161	Action 40	**82.31%**	**80.90%**	**81.73%**	**81.11%**
	CIFAR-10	91.01%	90.99%	90.98%	90.97%

Table 3.2: Classification metrics per model architecture and dataset.

Based on the top-ranked concepts extracted, we calculate the precision (Formula 3.2) between the images and their class in four different ways:

5L-1G : The top 5 local concepts for each instance and the top 1 global concept for each class.

5L-5G : The top 5 local concepts for each instance and the top 5 global concepts for each class.

5L-10G : The top 5 local concepts for each instance and the top 10 global concepts for each class.

10L-10G : The top 10 local concepts for each instance and the top 10 global concepts for each class.

The rationale behind varying the number of top concepts is to determine how many top global ranking concepts may best represent images from the same class.

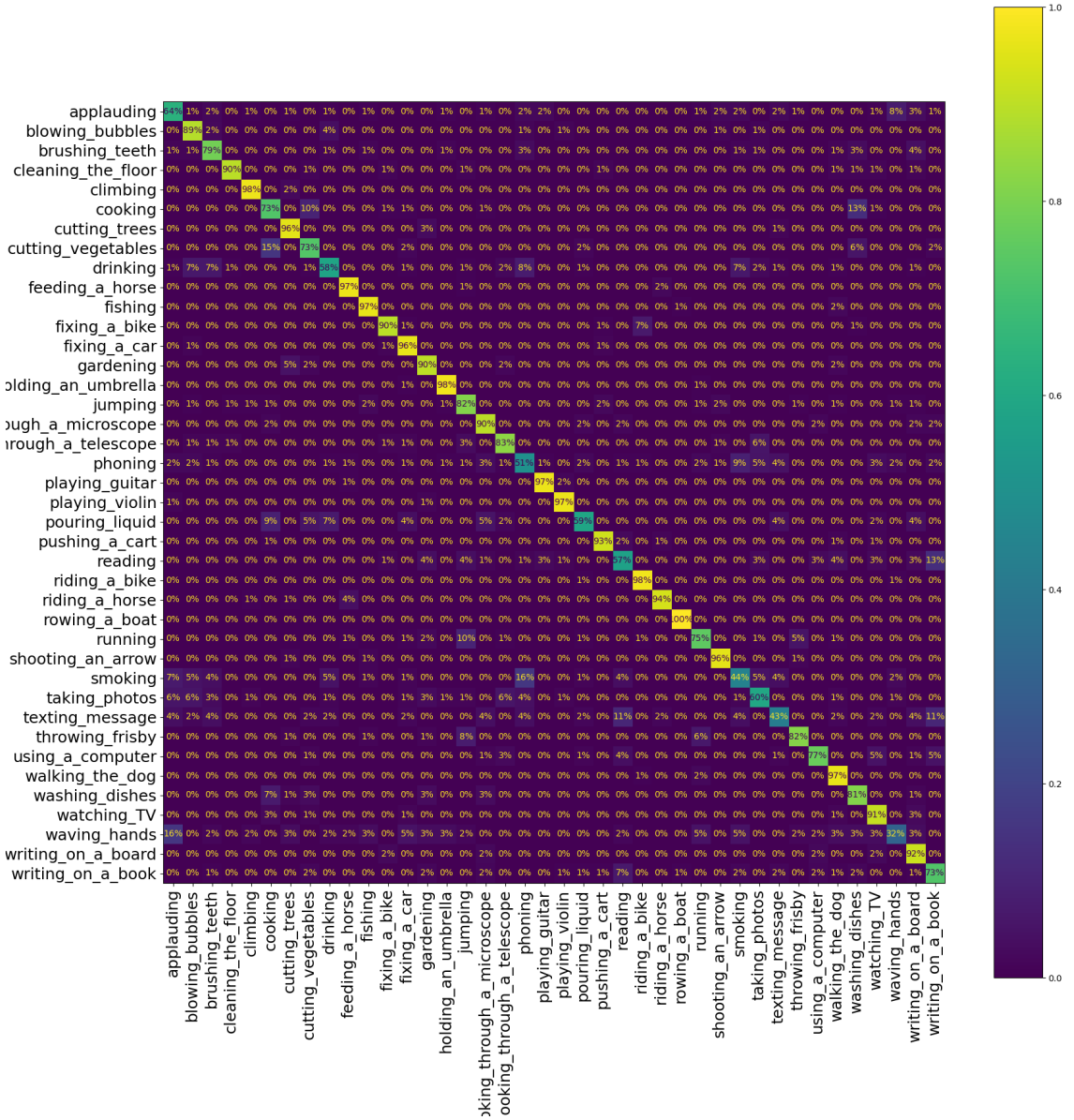

Figure 3.4: Confusion Matrix from SVM classifier on Action40, for ResNet-152 model output

The chosen number of concepts analysed (1, 5, and 10) was arbitrary for this experiment, but guided by the idea that three increasing values would be sufficient to show a trend, and that more than 10 concepts would be likely to pick up unrelated concepts for an image (hence 10 as the maximum value). With the minimum being set to 1 and the maximum being set to 10, we selected the value in the middle (5) as a third value. We observe that the linear classifier provides feature relevance

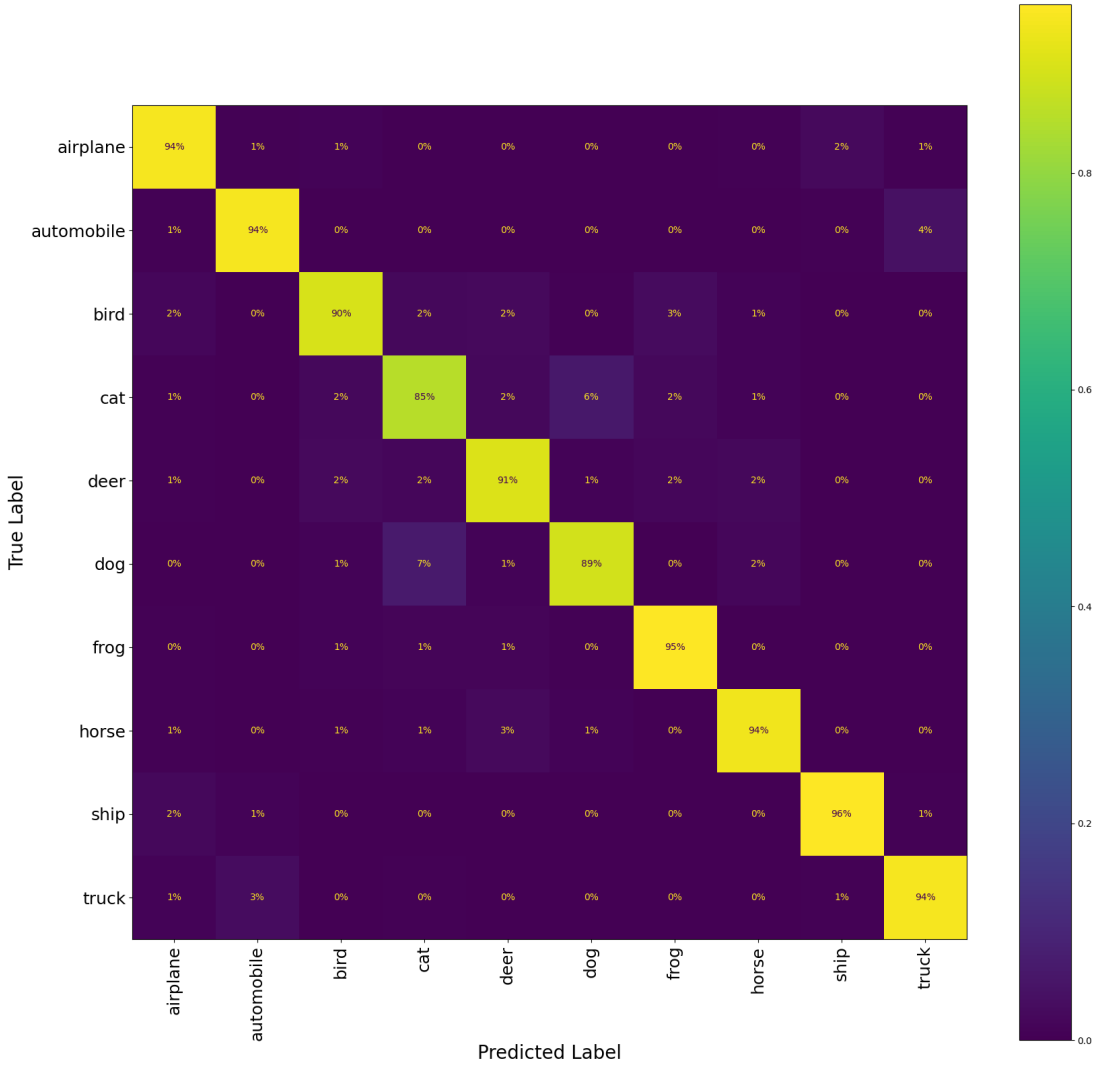

Figure 3.5: Confusion Matrix from SVM classifier on CIFAR-10, for ResNet-152 model output

depending on how successfully that feature separates the class. Relevant features thus identified for a class (globally) do not always correspond to top local concepts. For example, the class "phoning" in the *Action40* dataset has the most frequent top local concepts "person", "hand", "hair", "road", "bed", "plant", "apparel", "bus", "arm" and "leg" when we analyse the *ResNet-152* architecture (Appendix Table B.3). Analysing the top 10 global concepts for the same class, we have "screen",

"bicycle", "pool table", "food", "body", "bottle", "apparel", "pole" and "dog" (Appendix Table B.4). This means that in the model, the most common concepts identified for a group of images from the same class are not necessarily chosen as the best representative characteristics for that class. Looking at variations in the number of top concepts considered, we aim to find a threshold that increases precision between the local and global concepts of the same class.

3.4.2 Results

The *5L-1G* assessment considered whether the top global concept from a particular class was present among the top five local concepts. Figure 3.6 and 3.7 show that we were able to achieve a precision greater than 80% in only 6 out of 40 classes for *Action40* and 3 out of 10 for *CIFAR-10* by using just one top global concept with the *Resnet-152* architecture. This behaviour supports the previous intuition by emphasising that feature relevance in a linear classifier targets the characteristics that best distinguish (or separate) the classes.

When we examine the precision across the different thresholds for local and global concepts, we observe that using the top ten concepts in local and global instances (Figure 3.8 for *Action40* and Figure 3.9) within the same model achieves significant improvements in precision, indicating that the global top ten concepts are represented within the local top ten concepts. Given that the model identified 162 different semantic concepts and our technique could achieve a mean precision of 95% among only ten ranked concepts, this represents an important result. The mean precision and standard deviation for all classes, for each configuration varying the number of global and local concepts are shown in Figure 3.10 for the *Action40* dataset and Figure 3.11 for the *CIFAR-10* dataset, both for the *Resnet-152* architecture. The results for all approaches are presented in Table 3.3.

We must note that we only use precision as a quantitative measure for our concept ranking method at this stage. This is because we only check if global concepts are present among local ones. To illustrate our outcome qualitatively

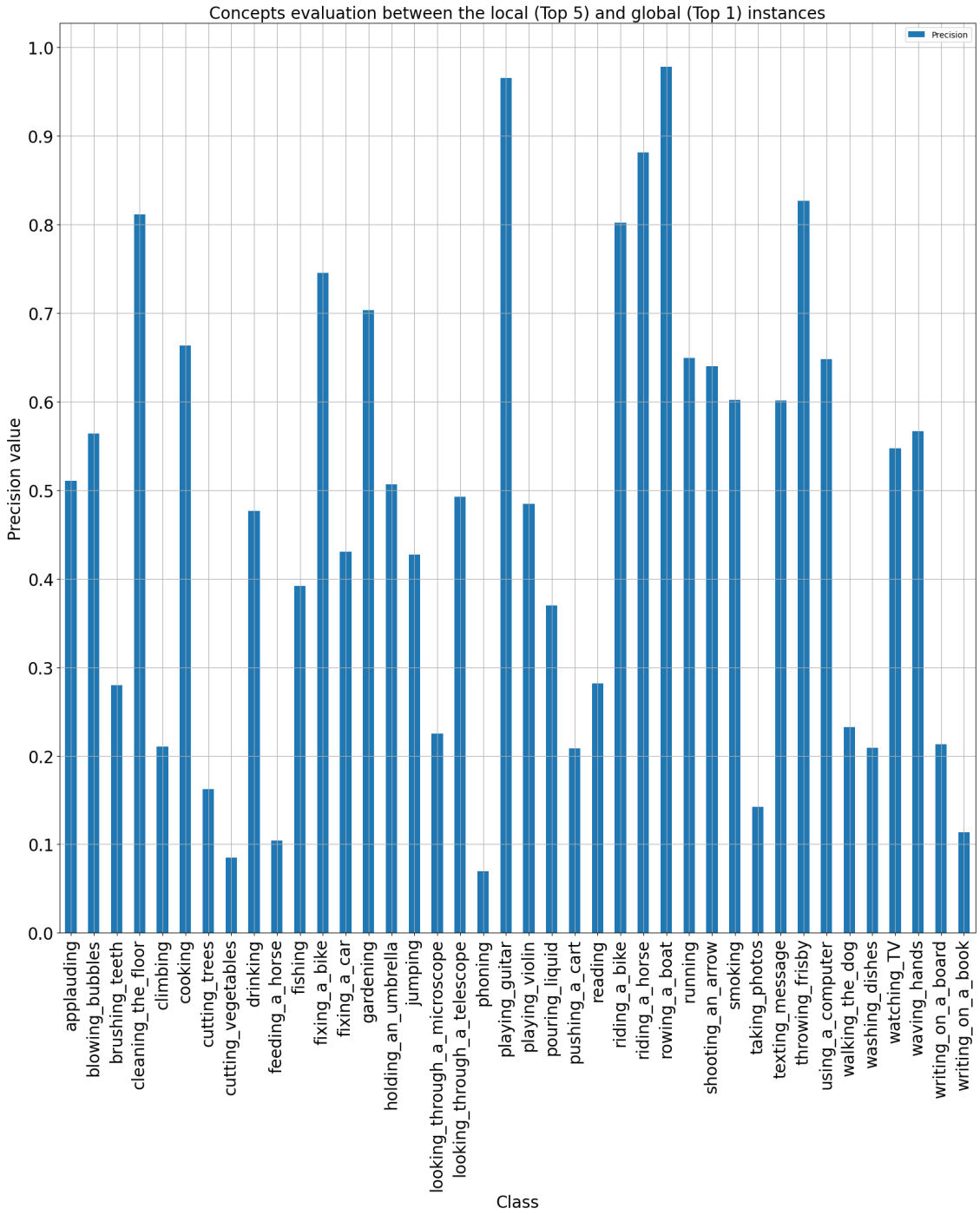

Figure 3.6: Top 5 local concepts X Top 1 Global concept - Action40 dataset and Resnet-152 model

with an example, let's look at "cutting_trees" (the class with the highest precision) for the *Action40* dataset and *ResNet-152* model. The class "cutting_trees" had 96% (Figure 3.4) of accuracy from the linear classifier and achieved 100% precision between the global and local concepts (Figure 3.8). Based on the feature significance from the linear model, the global concepts for this class are: "snow", "tree", "bird",

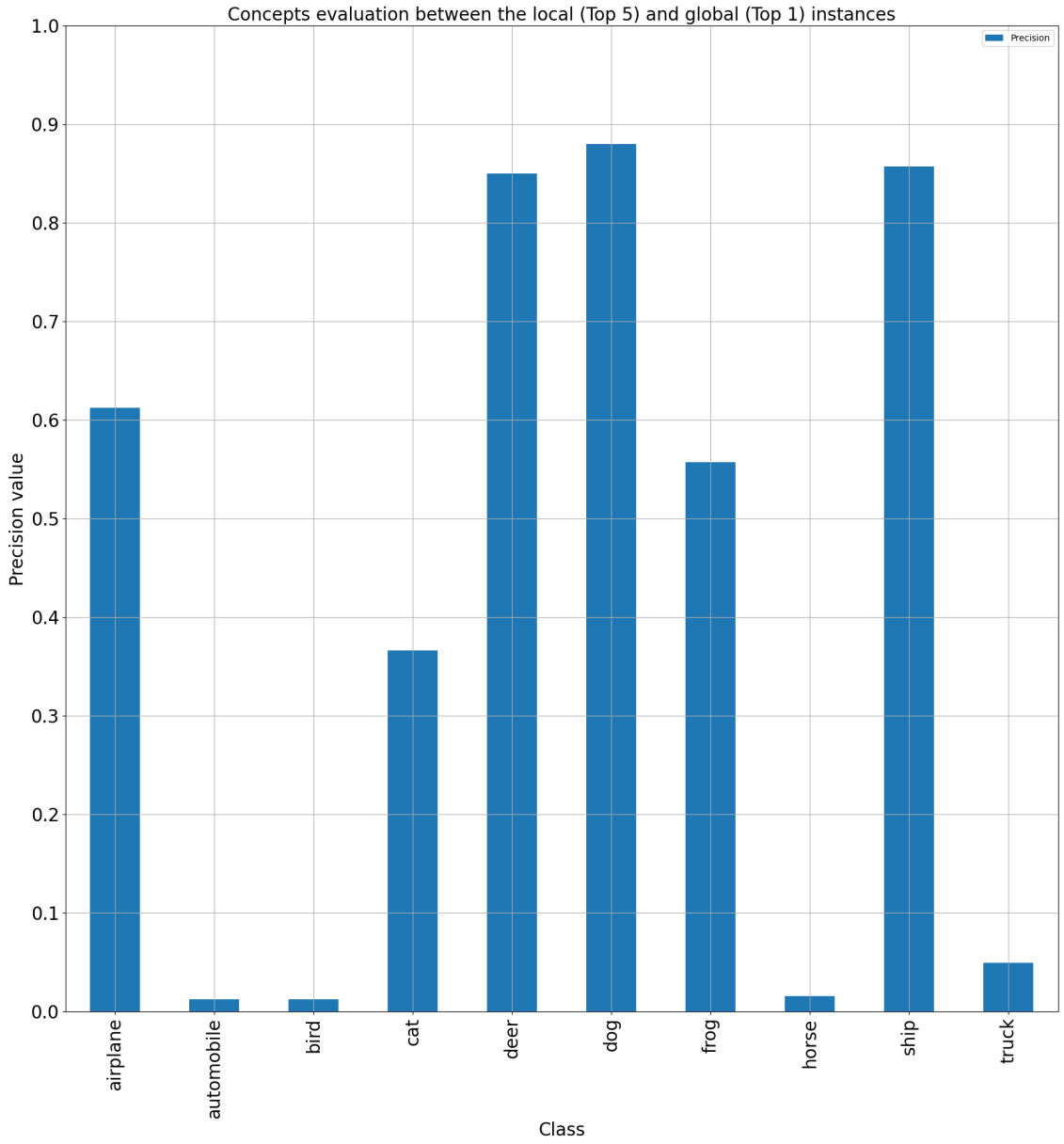

Figure 3.7: Top 5 local concepts X Top 1 Global concept - CIFAR-10 dataset and Resnet-152 model

"house", "plant", "motorbike", "hand" and "bicycle" (Appendix Table B.4). When we look at all of the images in the same class, the top ten local concepts are: "house", "tree", "plant", "bird", "person", "bicycle", "hand", "motorbike", "snow" and "food" (Appendix Table B.3).

This result demonstrates an overlap between global and local concepts for the class "cutting_trees", which we can use to describe what the model learned in terms of the concepts that best characterise this class as reflected in the local instances. We can manually check that this makes sense when we consider the activity of cutting

		Top 5 L X Top 1 G	Top 5 L X Top 5 G	Top 5 L X Top 10 G	Top 10 L X Top 10 G
ResNet-152	Action 40	47.05% (0.25)	80.10% (0.17)	87.59% (0.12)	**95.74**% (0.6)
	CIFAR-10	42.14% (0.37)	79.16% (0.15)	81.37% (0.14)	**95.88**% (0.03)
ResNet-50	Action 40	33.49% (0.29)	69.82% (0.21)	81.33% (0.16)	94.25% (0.07)
	CIFAR-10	34.33% (0.36)	69.01% (0.35)	78.55% (0.29)	95.08% (0.08)
DenseNet-161	Action 40	36.33% (0.26)	72.80% (0.18)	84.39% (0.11)	95.53% (0.05)
	CIFAR-10	28.53% (0.33)	63.22% (0.25)	73.19% (0.22)	92.44% (0.08)

Table 3.3: Precision mean and Standard Deviation between the approaches, by model and dataset.

the tree and its images in the dataset. As we stated, this is merely an intuition, and a more systematic evaluation (either conducted manually by humans or automatically via labels) should be performed to validate this claim more broadly. When we look at the "phoning" class, the linear classifier did not produce good results (precision of 51%), and when we look at the global and local concepts, the result was the lowest (about 67%). This outcome might indicate three possibilities: the linear model did not separate the concepts properly (an issue with the linear model); there is a lack of concepts that could better describe this class (an issue with concept generation); or could be an issue with bias in the data.

To summarise, our experimental analysis so far has shown that the first framework's component was able to successfully retrieve the top ten concepts from disentangled representations that best characterise the local instances (as per *Network Dissection*) as well as the global instances, answering **RQ1** (Section 1.5). We assessed feature importance by comparing the existence of concepts in local and global occurrences for **RQ2** (Section 1.5). In this first component, we presented the potential of using disentangled representations to provide a semantically meaningful interpretation of classification results produced by a *CNN* regarding relevant semantic concepts. By using a linear classifier such as *SVM*, we can meaningfully rank

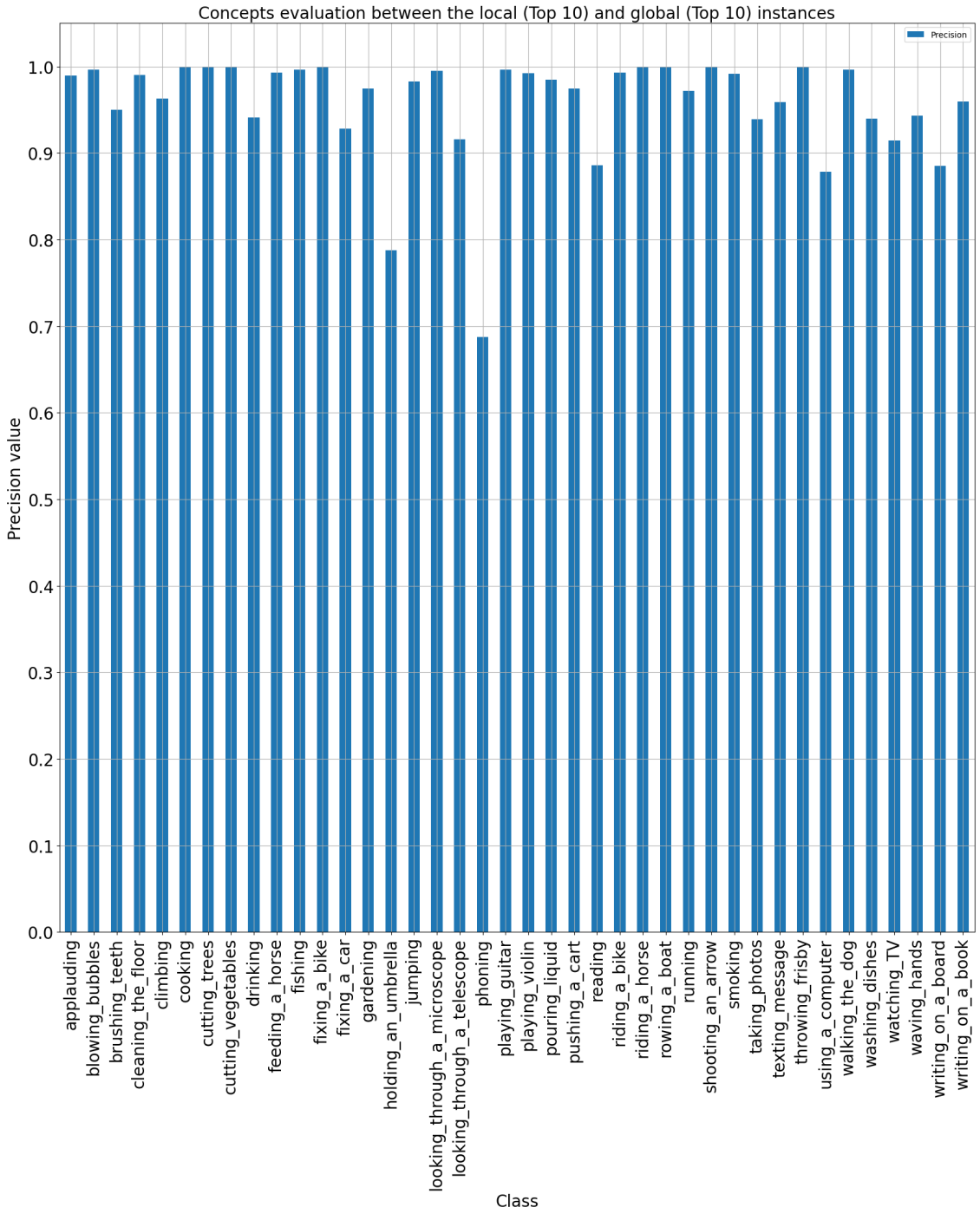

Figure 3.8: Top 10 local concepts X Top 10 Global concept Action40 dataset and Resnet-152 model

the top ten concepts that characterise an instance and, more generally, a class from locally disentangled representations.

We demonstrated that we can identify the top concepts for an image of a given class and that these are the same concepts required to best separate this class. For example, with a precision of 95% between the concepts presented in the images and

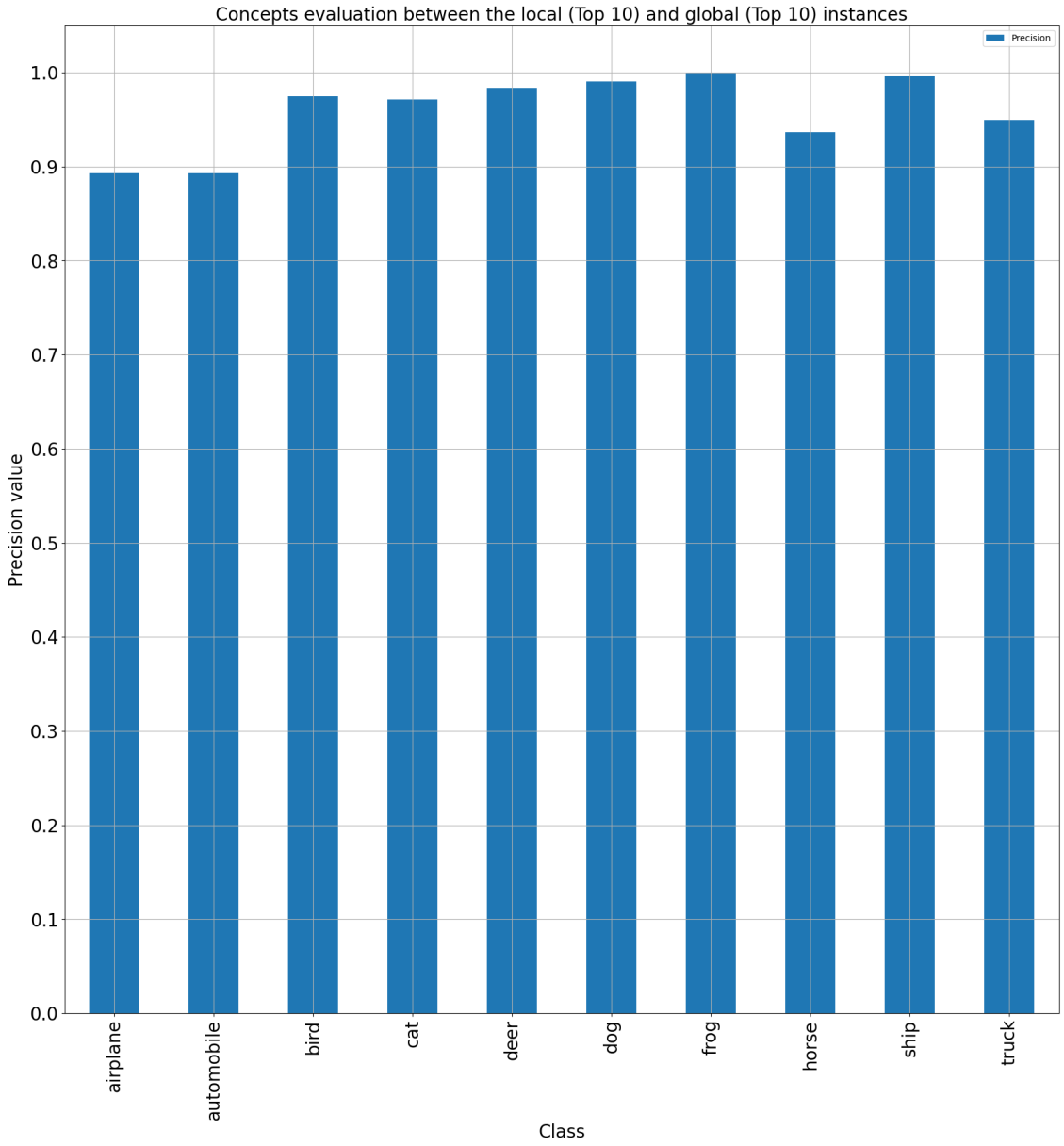

Figure 3.9: Top 10 local concepts X Top 10 Global concept CIFAR-10 dataset and Resnet-152 model

their class, we find that images categorised as "riding a bike" contain the top local concepts "bicycle" and "wheel", and the same top concepts are necessary to separate this class according to the linear classifier. As a result, we argue that the model has learned those concepts related to the specific class (and instances of that class). This paves the way for a concept-driven explanation of classification results using disentangled representations, although there are several limitations to this approach that we are going to explore in the next Chapter of this thesis.

For example, the fact that this component relying on concept extraction alone

cannot identify semantic relations between extracted concepts, which can substantially enhance the quality of an explanation. To this end, leveraging an external knowledge base can help detect semantic relations between those concepts. Hence, the second component of our framework (discussed in Chapter 4) aims to tackle this challenge, providing a further step towards a broader transparency and better understanding of what the model has learned. Another key challenge is that the human expert's ability to semantically explain outcomes is limited due to a lack of information regarding causal links between concepts and their relations. To tackle this issue, we developed the third component of the framework (Chapter 5), offering the possibility of leveraging the extracted concepts and relations to learn symbolic deduction rules representing the network's behaviour.

3.5 Summary

This Chapter introduces the first component of our framework, aimed at understanding the deep representations of a trained Convolutional Neural Network (*CNN*) in terms of semantic concepts. The component focuses on extracting human-understandable concepts from the model, identifying relevant local concepts for individual images and global concepts for an entire class. Existing interpretation techniques, primarily visual and attention-based, often fail to represent image concepts across similar images consistently. To address these limitations, this framework relies on disentangled representations for concept extraction and linear models for ranking of semantic concepts.

Specifically, the concept extraction process leverages the *Network Dissection* technique, which identifies how individual hidden units in the *CNN* align with semantic concepts such as object, part, material, colour, texture, and scene. Using three architectures (*ResNet-152* and *DenseNet-161*), known for identifying the most unique concepts [141], and *ResNet-50* to verify whether a smaller *ResNet* model would yield comparable results, the identified concepts are then connected to output classes and individual images. The method focuses on the last *CNN* layer to

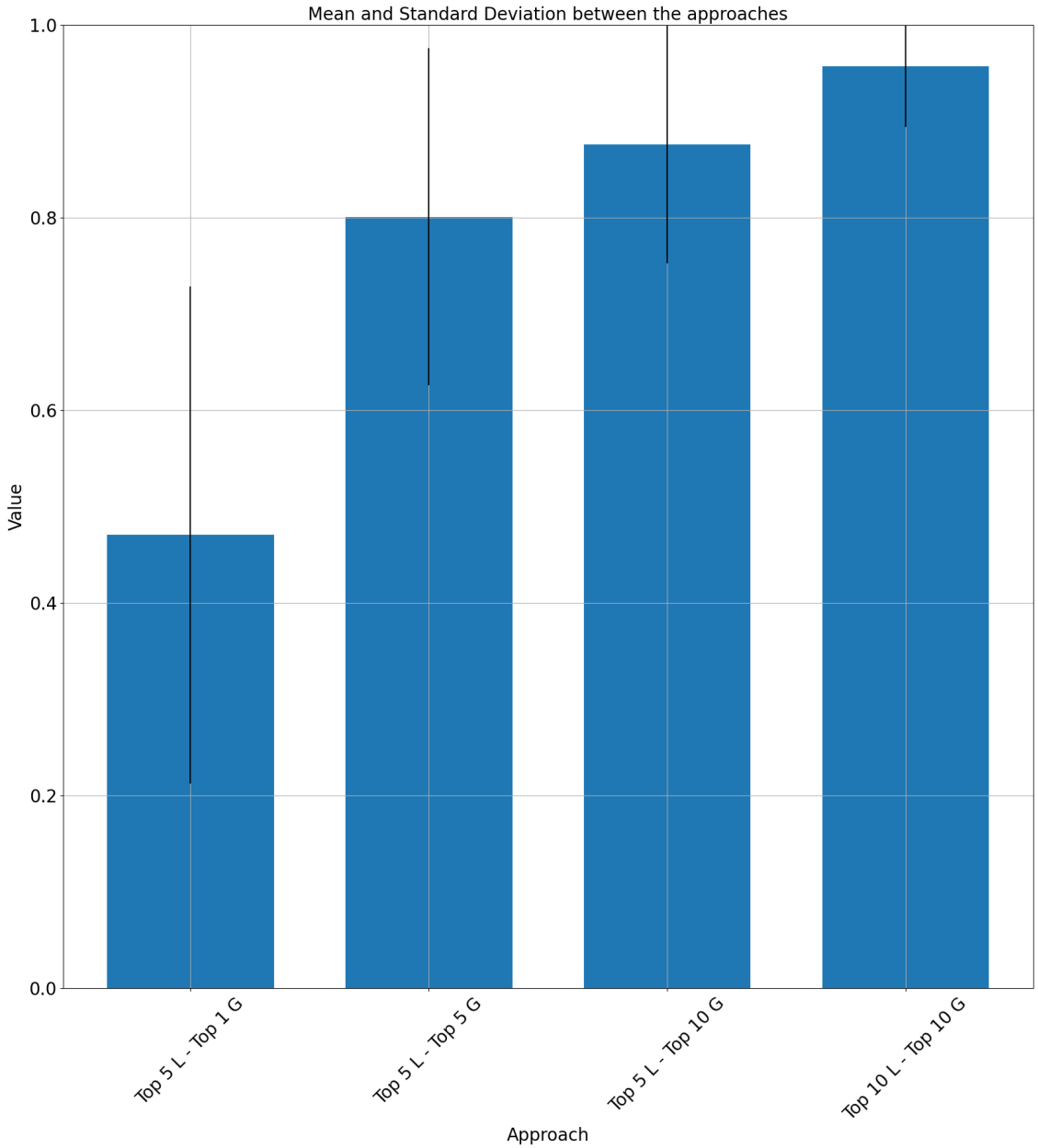

Figure 3.10: Mean and Standard Deviation between the Precision - Action40 dataset and Resnet-152 model

maximise the discovery of unique, high-level semantic concepts, thereby providing a more comprehensive understanding of the semantics the model has learned from low level features.

To evaluate the importance of these concepts, the framework uses a precision-based metric that compares the ranked local concepts (from individual images) with the ranked global concepts (for entire classes). The experiment, conducted using the *Action40* and *CIFAR-10* datasets, provides encouraging results. Using a

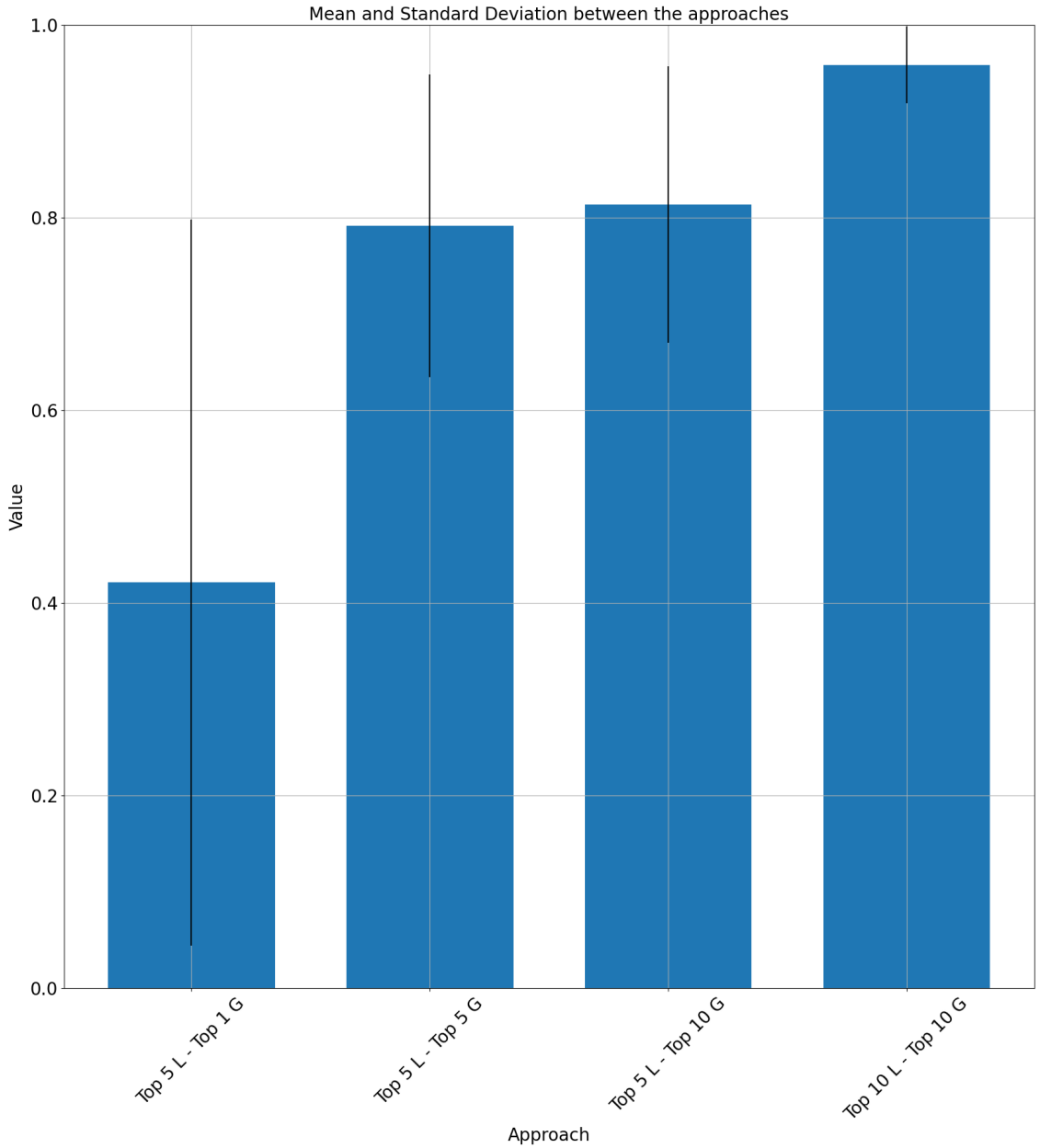

Figure 3.11: Mean and Standard Deviation between the Precision - CIFAR-10 dataset and Resnet-152 model

transfer learning method, the framework adapts the identified concepts to the new dataset, and an *SVM* linear classifier ranks these global concepts. The results show a significant overlap between local and global concepts, with a mean precision of 95% on *Action40*, indicating that the model's top concepts for an image are consistent with those necessary to distinguish its class.

In conclusion, the work in this Chapter demonstrates the potential of using disentangled representations to provide semantically meaningful interpretations of

CNN classification results. The framework's ability to retrieve and rank relevant concepts lays the foundation for concept-driven explanations.

Chapter 4

Extracting Relations

4.1 Introduction

This Chapter introduces the second component of our framework, which builds upon our concept extraction component in order to elicit relations between concepts which are likely to have been learned by the *CNN*. To achieve this, we propose a method that relies on using external knowledge. This approach is expected to enhance interpretability of *CNN* models in image classification tasks, as it allows to go one step further in the semantic understanding of the model's inner working by relating concepts with each other with semantic relations relevant for a specific class. Moving beyond concepts to identify relations is an important step for explainability, since we as humans understand and explain deduction processes using relations (causal, spatial, functional, among others) .

We can define a relation as a connection or association between two or more entities, concepts, or variables. This connection is based on factors including causality, similarity, proximity, or functional dependence. For example, the concept "wheel" relates to the concept "bike" through the relation "is part of". Generally, this relation is inherently assumed when these two concepts are present in the same image, but there is no visual counterpart that can be immediately associated to the presence (or absence) of the relation. Based on this idea, we believe that external knowledge is a necessary resource to enable such relations to be identified, and we

have identified specific knowledge graphs for this purpose.

A knowledge graph (Section 1.3) provides a structured representation of entities and their relations, allowing for a deeper understanding of the context in which the concepts operate. This contextual understanding is essential for explaining the decisions made by the model, as it enables the identification of relevant factors and their interactions [53]. Knowledge graphs can also help disambiguate entities by linking them to specific concepts and relations. This is particularly important in the context of *CNNs*, where entities may be represented by multiple features or attributes. Additionally, knowledge graphs can model complex relations between entities, which is critical for understanding the decision-making processes of the model.

As previously mentioned in Section 2.2.1, the *Common-sense Knowledge Graph (CSKG)* [57] includes a set of seven popular sources[1] to build a consolidated representation as relational graph, focused on common relations between objects. One example is depicted in Figure 4.1, where we can see on the left side an image[2] representing the object "piano" and on the right, the object representation in the knowledge graph with its relations in *CSKG*. This graph was created by selecting the main concept, "piano", and gathering the concepts and relations directly connected to it [57]. In this knowledge graph piece, we can see that the object "keys" has a relation "Part Of" with the object "piano", which has a relation "Used For" with the object "music". Looking at the image, we can identify most of the relations regarding the object as presented in the knowledge graph and how other objects interact. Our general assumption is that a person can comprehend these interactions with minimal prior knowledge, hence presenting a similar description of how the model's concepts are related can enhance human understanding of what the model learned.

Bearing in mind the advantage of being able to identify relations among concepts for enhancing human understanding, the main contributions of this Chapter

[1] https://github.com/usc-isi-i2/cskg
[2] https://depositphotos.com/photo/woman-playing-piano-121705144.html

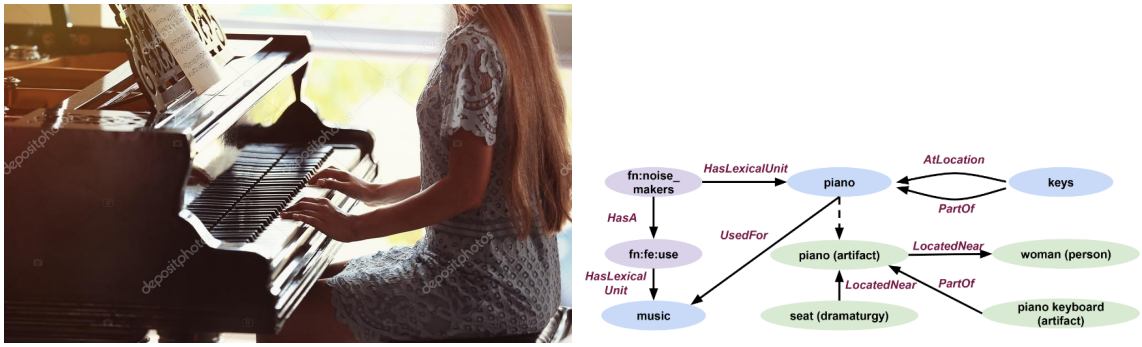

Figure 4.1: A Piano image example (left) and the Piano relations on *CSKG* (right)

are: (i) an approach which leverages a common-sense external knowledge graph and combines it with the concepts extracted from the first component of the framework (Section 3.2), to identify candidate relations learned by the model **(RQ3)** (Section 1.5); (ii) an evaluation method relying on a multimodal dataset (the Validation dataset), focused on verifying whether the relations among extracted concepts were in fact learned by the model **(RQ4)** (Section 1.5). Both contributions are represented in Figure 4.2. In this component, we use the concepts extracted from the previous step to gather candidate relations from a knowledge graph (*CSKG*). These relations are identified based on string matches between pairs of concepts and the corresponding nodes within the graph, and the relations directly linking each concept pair. To validate the relations learned by the model, we rely on a validation dataset (*Visual Genome*), which contains images labelled with relations. We selected only images that represent the relation candidates, and then use them to evaluate whether the relation was learned or not. Further details about this process will be explored in the following sections.

To the best of our knowledge, this component represents a novel approach to understanding deep relation representation and can be extended by employing different external knowledge sources and representations. Alongside this process, the formula employed to verify the significance of the relation showcases an innovative aspect of the work. The implementation can be found online[3].

[3]https://github.com/EricFerreiraS/relation_extraction_AICS24

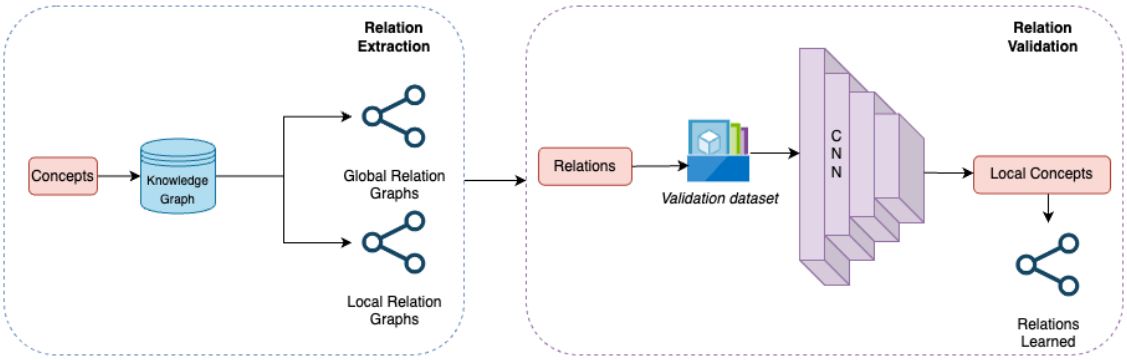

Figure 4.2: Relations Extraction Component

4.2 Relation Extraction (RQ3)

Given a knowledge graph $G = (V, E)$, where $V = \{v_0, v_1, ..., v_n\}$ is the set of n nodes that represent concepts and E is the set of relations (edges) $e_{ij} = (v_i, v_j)$ between pairs of concepts v_i and v_j, we are going to use the set of top 10 local and top 10 global concepts extracted in Section 3.2 to select the V nodes in the graph.

We only retrieve direct relations between concepts, which implies returning only direct edges between each pair of concept nodes (v_i, v_j), such that:

$$Distance(v_i, v_j) = 1 \tag{4.1}$$

We have applied this restriction in order to limit the excessive noise derived from extracting multi-hop relations that might not be relevant for a particular class, but also reduce the combinatorial complexity. One limitation of this direct string matching is that we do not consider syntactic variance or synonyms of concepts that are slightly more general or more specific with respect to the concept name under consideration. This could result in excluding semantic relations that might be relevant to enhance the model's understanding. This issue could be addressed with an additional step that extends the initial set of concepts from the first step by exploring parts of the Knowledge Graph around the concept, e.g. by exploring subclass and type-of relations. We leave this to future work.

The Common-sense Knowledge Graph *CSKG* [57] has labelled 34 distinct edge relations. A very common and important relation is the "is a" relation (which we

could consider as a subsumption relation), meaning that for $e_{ij} = (v_i, v_j)$, if v_i is a sub-type or instance of v_j, then every v_i *is a* v_j. As an example, if you consider $v_i = car$ and $v_j = vehicle$, there is an edge for the subsumption relation given by the fact that every "car" is a "vehicle" in *CKSG*. The full list of all common-sense relations can be found here[4], which is shared between *CSKG* and *Conceptnet*.

In the context of our framework, this step takes as input the outcome from the concept extraction component (Section 3.2): the top 10 relevant local and global concepts. These concepts are looked up in terms of their representation as nodes in the knowledge graph, and all the edges directly linking any pair of concepts is also retrieved. For example, given a set of concepts c_1, c_2, c_3 as the most relevant concepts for an image x_i, the method associates each concept to a node in the knowledge graph v_1, v_2, v_3 and returns any edge e_{ij} which links only these nodes directly, without considering other nodes that may have edges in the graph which are not relevant for x_i. This way, each retrieved edge would represent a direct relation between the concepts learned in the first part of the framework. The gathered correlations between the concepts do not represent the fact that the model has learned them, as at this stage they are candidate relations with respect to *CSKG*; as such, these relations refer to all the common-sense knowledge of possible direct relations among the concepts. Once these candidate relations have been identified, they still need to be validated to determine whether we have evidence to believe they have been learned by the model.

It is important to note that matching concepts and nodes in the knowledge graph relies solely on string matching, as each identified concept corresponds to a node in the graph. Depending on the scenario, it may be relevant to consider employing syntactic variations in the names of the concepts, which could encompass a broader range of nodes and, consequently, yield additional relations. Since this thesis focuses on a general domain and defines the base strategy, we relied entirely on string matching in the current implementation.

[4]https://github.com/commonsense/conceptnet5/wiki/Relations

4.3 Relation Evaluation (RQ4)

In order to validate candidate semantic relations our approach relies on *Visual Genome* [65], a dataset that connects structured image concepts to language via textual labels. This dataset contains pictures with tagged relations separated by bounding boxes. An example from this dataset can be seen in Figure 4.3, where the relation between two objects, such as "man" and "shirt", is highlighted by a bounding box. Each image may contain more than one relation (possibly overlapping), and the relations' name are different from the knowledge graph labels.

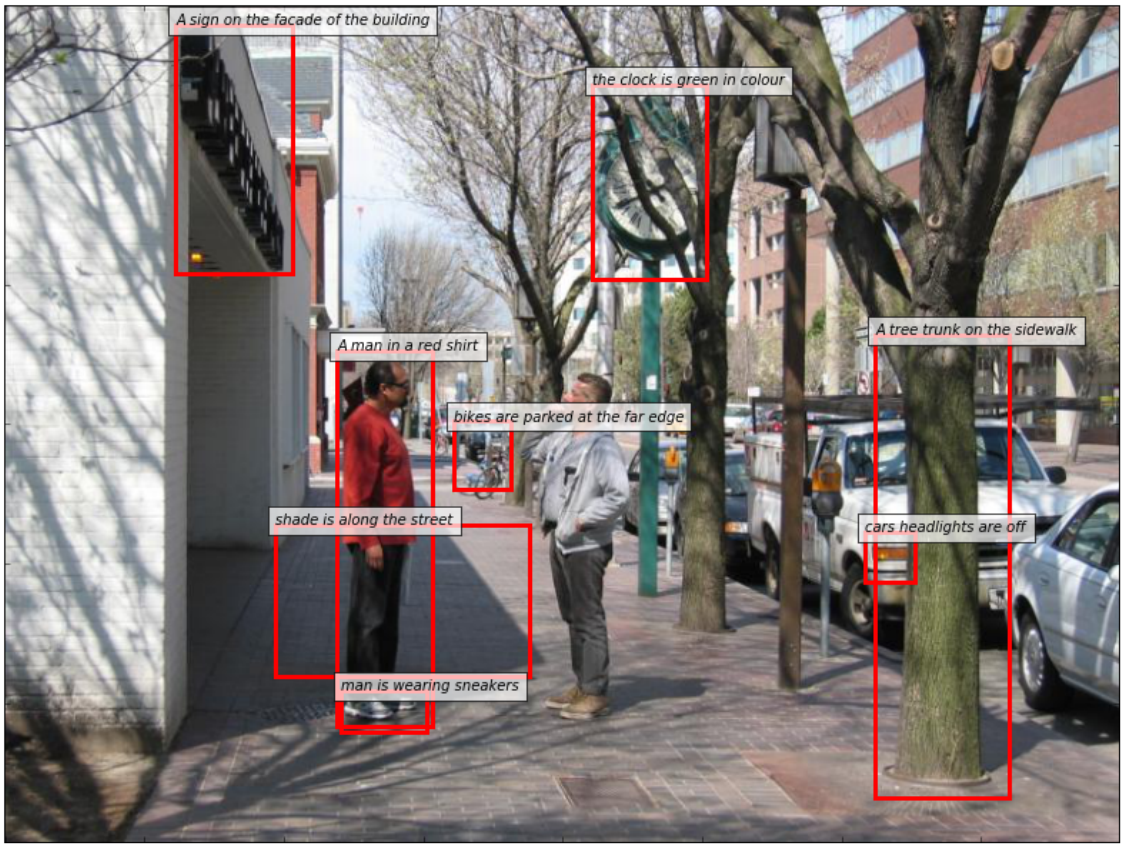

Figure 4.3: Visual Genome [65] example

In order to reduce the number of images from *Visual Genome* that will be used and focus on the images that are more likely relevant for the validation process, we only considered images containing at least two concepts annotated in the dataset meta-data, from the top 10 local and global concepts identified by the concept extraction component. Consider C as the set of the top ten global and local concepts c_i returned by the concept extraction component (Section 3.2). Once the images

relevant to the set of concepts in C are identified, we create one image per cropped relation with the corresponding relation as label. The set of all these cropped images annotated with relation names represents our validation dataset.

The validation process is as follows:

1. for every relation r_i among concepts c_1 and c_2 extracted from *CSKG*, we identify all images I_j in *Visual Genome* representing r_i[5];

2. every image I_j for a given relation r_i is passed through the model used for concept detection, with the hypothesis that if the two relevant concepts c_1 and c_2 for r_i are among the top ten activated filters, the network has likely learned the corresponding relations;

3. due to differences in relation names between *CSKG* and *Visual Genome*, we relaxed the exact string matching approach by comparing the use of Named-Entity Resolution (*NER*) approaches from Natural Language Processing (namely from *spacy.io*[6] and *NLTK*[7]) and the well known *word2vec*[8].

Thus, if the concepts "bicycle" and "wheel" are activated in two different filters, and the same filters are activated for an image from *Visual Genome* which contains the relation "wheel is part of a bicycle", and these two filters are both in the most highlighted concepts together, locally or globally, the relation between them was learned. With a set of learned relations R, we use it to validate the relation candidates resulting from the relation extraction step.

The reason we selected *Visual Genome* as a more robust approach to validation as opposed to Large Language Models such as *GPT-4*[9] or *Gemini*[10], is the reduced risk of hallucinations [47], as well as the ability to have access to a visual representation of relations that we could use as a ground-truth to relate concepts to disentangled filters. In Chapter 5, we explore an alternative approach for extracting concepts

[5]Here we use string matching between *CSKG* and *Visual Genome* concepts
[6]https://spacy.io/models/en#en_core_web_lg
[7]https://www.nltk.org/index.html
[8]https://radimrehurek.com/gensim/models/word2vec.html
[9]https://openai.com/index/gpt-4/
[10]https://gemini.google.com/app

and relations. This method employs a multimodal model to identify concepts and relations rather than relying on network dissection and knowledge graphs. However, this approach has the potential to introduce more noise and is more challenging to control; consequently, this thesis does not include a systematic comparison of the two strategies.

We believe the ability to extract high-level human concepts and their relations has potential and can be used to generate simple explanations for image classification tasks performed by a *CNN* in terms of the relations between concepts. In the next Section, we present an experiment with this method.

4.4 Experimental Setup

Our experiments were conducted on a machine running Linux Mint 21.2, with 48 CPUs and 128 GB of RAM. We rely on two main Python libraries: Pytorch[11] for training and testing the models and *KGTK*[12] to work with the knowledge graph. We used as input in this step the concepts extracted from the last layer of the *ResNet-152* trained model on *ImageNet*, as we presented in Chapter 3.

The concepts extraction phase on *ResNet-152* was pre-trained on the *Imagenet* dataset, and it resulted in 162 different concepts including object, part, material, and colour. We started from this pre-trained model and applied transfer learning, freezing all the trained layers and replacing the fully connected layers with a linear classifier (*SVM* kernel). This enabled us to perform relation extraction and validation (with string matching and with Named-Entity Resolution or *NER*) on two different datasets, namely *Action40* and *CIFAR-10* (Appendix B.1) as discussed in the remainder of this Section.

Extracting candidate relations. Using the *CSKG* knowledge graph, we search for the common-sense relations between the concepts discovered in the concept extraction phase. For each (global and local) concept pair we look for a direct

[11] https://pytorch.org/get-started/locally/
[12] https://kgtk.readthedocs.io/en/latest/

Figure 4.4: Image example from *Action40* dataset - Class "Smoking"

connection in the common-sense knowledge graph *CSKG*. For example, Figure 4.4 represents an image from the class "smoking" in the *Action40* dataset, which includes the top 10 local concepts: "hand", "person", "dog", "stage", "bed", "seat", "bottle", "bathtub", "airplane", and "car". The top 10 concepts for the class are "hand", "plate", "airplane", "screen", "hair", "person", "concrete", "horse", and "car"; both sets were extracted in the first component of the framework.

Using these concepts, we search within *CSKG* for the candidates that comprise a direct relation between the pair of concepts. For this example, the local relation candidates are: "person related to bed", "bed synonym seat", "bed located in car", "bed located behind person", "car has a seat", among others; similarly, the global relation candidates are: "car parked on concrete", "person in front of a plate", "horse standing on concrete", among others. Extending this process for all the images in the dataset, we gather the local and global candidate relations for the entire dataset.

Furthermore, based on the local and global concepts extracted from the first

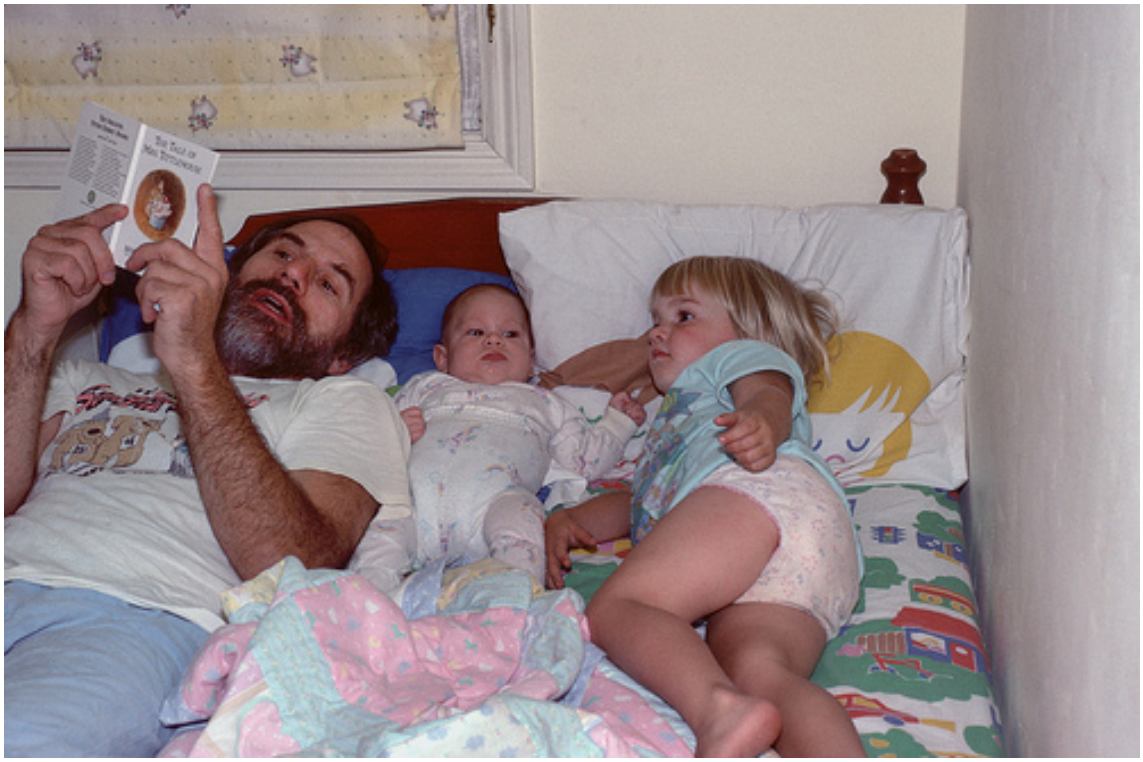

Figure 4.5: Person laying on a bed - *Visual Genome* example

component, we also select images from *Visual Genome*, which can be used to validate our approach. Continuing with the example cited above (Figure 4.4), we search in *Visual Genome* for images that contain at least a pair of concepts (global or local), such as "person" and "bed" for instance, separating the image and the relation represented in the image for the validation step. Figure 4.5 illustrates one image from the *Visual Genome* which contains both concepts, "person" and "bed", and the relation "person laying on a bed".

The results of our analysis are presented in Table 4.1, where we can see the number of global (# R (Global)) and local (# R (Local)) unique relations extracted using *CSKG*, the percentage of the *Visual Genome* images containing the K local and global concepts extracted from each dataset (% VG Images), and the total number of unique relations learned using *Visual Genome* (#R (VG)). Note that the percentage of images from *Visual Genome*, and the relations learned are not distinguished as local and global, as they relate to the entire model when considering the overlap between local and global.

	# R (Global) CSKG	# R (Local) CSKG	% VG Images	# R (VG)
Action40	339	2,495	12%	3,434
CIFAR-10	137	2,176	14%	2,566

Table 4.1: Candidate local and global relations from *CSKG*, Percentage of the Visual Genome images that contain the same pair of concepts in local and global relations from *CSKG*, and Relations found using the *Visual Genome* only.

Validation of Candidate Relations. Based on the relations learned (Table 4.1, last column), we find relations that are present in both *CSKG* and *VG*, both local and global. Table 4.2 shows the number of common relations found by simple string matching (#R) and by relaxing the matching using *NER* approaches (#R_NER). We consider the sum of unique relations extracted by any of the three *NER* approaches. We did not consider the contribution of each single approach for this analysis. Table 4.2 also presents the percentage of relations validated from *Visual Genome* more than once, that were also present in *CSKG* (%R_VAL) separated in global and local relations respectively. We arbitrary use this threshold to increase the acceptance of the relation was learned. This means, for example, that starting from the candidate global relations (Table 4.1 #R (Global)), extracted from the combination of the top K features and the *CSKG* in the candidate relation extraction phase, only 8% were validated as learned by the model for the *Action40* dataset.

	# R (Global)	# R_NER (Global)	# R (Local)	# R_NER (Local)	% R_VAL (Global)	% R_VAL (Local)
Action40	7	51	23	166	8%	9%
CIFAR-10	1	15	23	118	2%	3%

Table 4.2: The number of relations found, locally and globally (R), their relaxation with NER (R_NER), and the percentage of global and local relations candidates validated.

Figures 4.8 and 4.9 present the number of unique relations in global and local concepts, respectively for the *Action40* dataset.

Illustrative Example. In order to understand the difference between finding *candidate* relation and validating them, we have two examples that illustrate them. For instance, based on the top 10 local concepts from Figure 4.6 - "motorbike", "pot", "wheel", "plant", "bicycle", "case", "metal", "person", "sidewalk", and "hand" - we found 30 relation candidates when we gathered the direct relations between the concepts from *CSKG*, such as "bicycle is made of metal", "person has a hand", and "person next to a motorbike", among others. After the validation step, only two relations were recognised as having been learned by the model - "person has a hand" and "wheel is part of a bicycle".

Figure 4.6: Fixing a Car Image [131]

Another example is illustrated by Figure 4.7, which shows "arm", "hand", "chair", "hair", "plant", "airplane", "person", "shop window", "bed", and "bus" as the top 10 local features extracted from the first component. It returned 30 relation candidates, such as "arm is part of a chair", "hand is part of an arm", and "chair is next to a person", among others. After validation, only three relations remain as learned

by the model: "arm is part of a char", "arm is part of a person", and "person has a hand". Both examples illustrate how the relation candidates, although feasible since they rely on an external source, need to be validated to confirm whether the model has learned them.

Figure 4.7: Applauding [131]

Measuring how well relations separate classes with Coverage. Overall, we observe that the there is a high number of local relations, but when validated across the instances of a class, only a few of those relations are likely to influence the classification task. Our method allows us to identify such global relations reducing noise.

In order to measure this phenomenon per class, we define the notion of coverage which measures how many local relations (for all images of a given class) are also present as global relations for that class. These global relations are relations among concepts specific to that class as they best separate that class's feature space. If

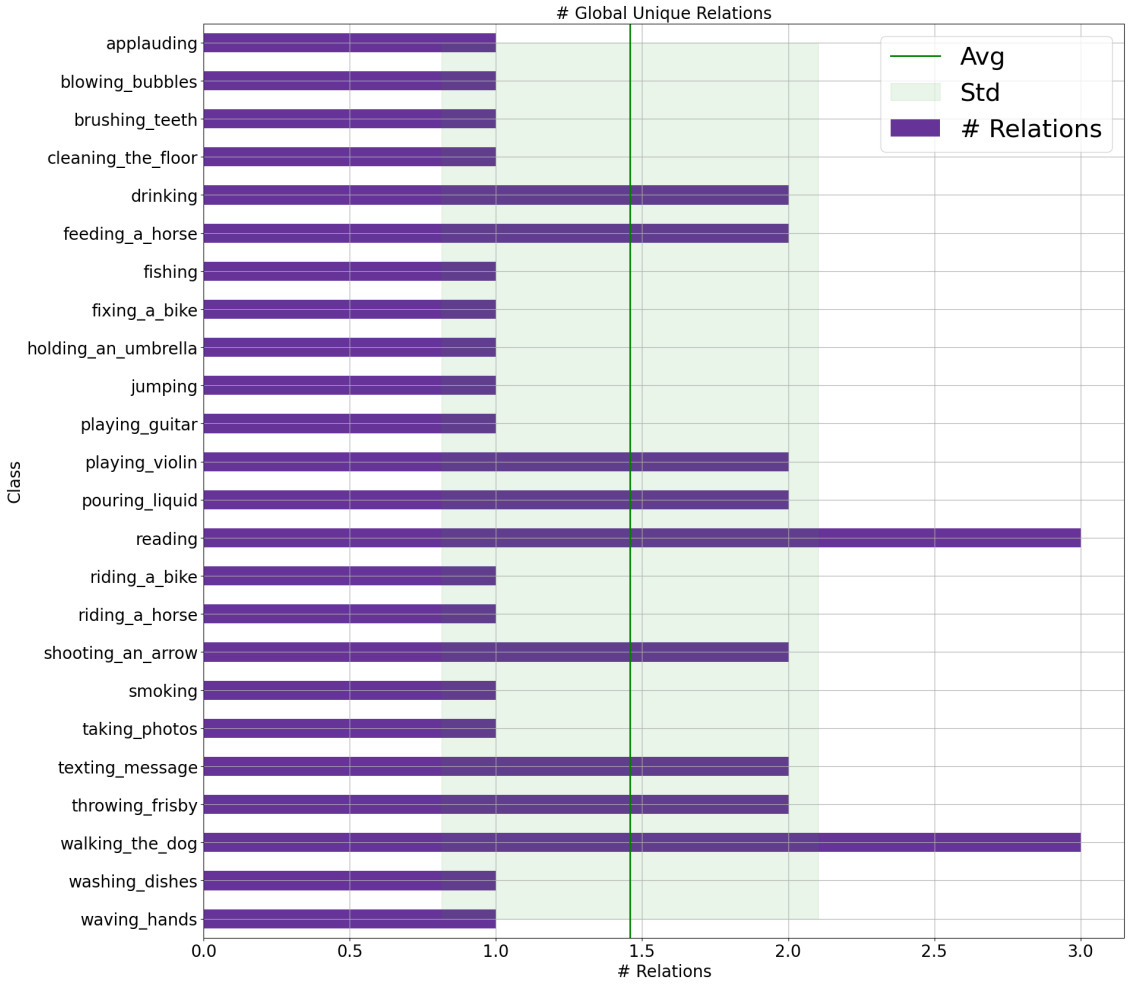

Figure 4.8: Number of Unique Global relations - Action40/ResNet-152

at least one relation is given, the globally rated relations make sense with the local ones. The coverage formula is as follows:

$$Coverage_c = \frac{\sum R_{l_c,g_c}}{\#L_c} \tag{4.2}$$

where $Coverage_c$ is the coverage for a specific class c, $\sum R_{l_c,g_c}$ is the sum of the instances where the local (l_c) and global (g_c) relations have at least one element in common for class c, and $\#L_c$ is the number of local instances that belongs to the class c. As expected, the intuition behind this formula closely resembles equation 3.2 presented in Section 3.3 of the previous chapter. Figures 4.10 and 4.11 present values of $Coverage_c$ for each of the two datasets.

This analysis helps clearly identify classes with very low coverage (such as "reading") where the identified local relations are not likely to be influencing the clas-

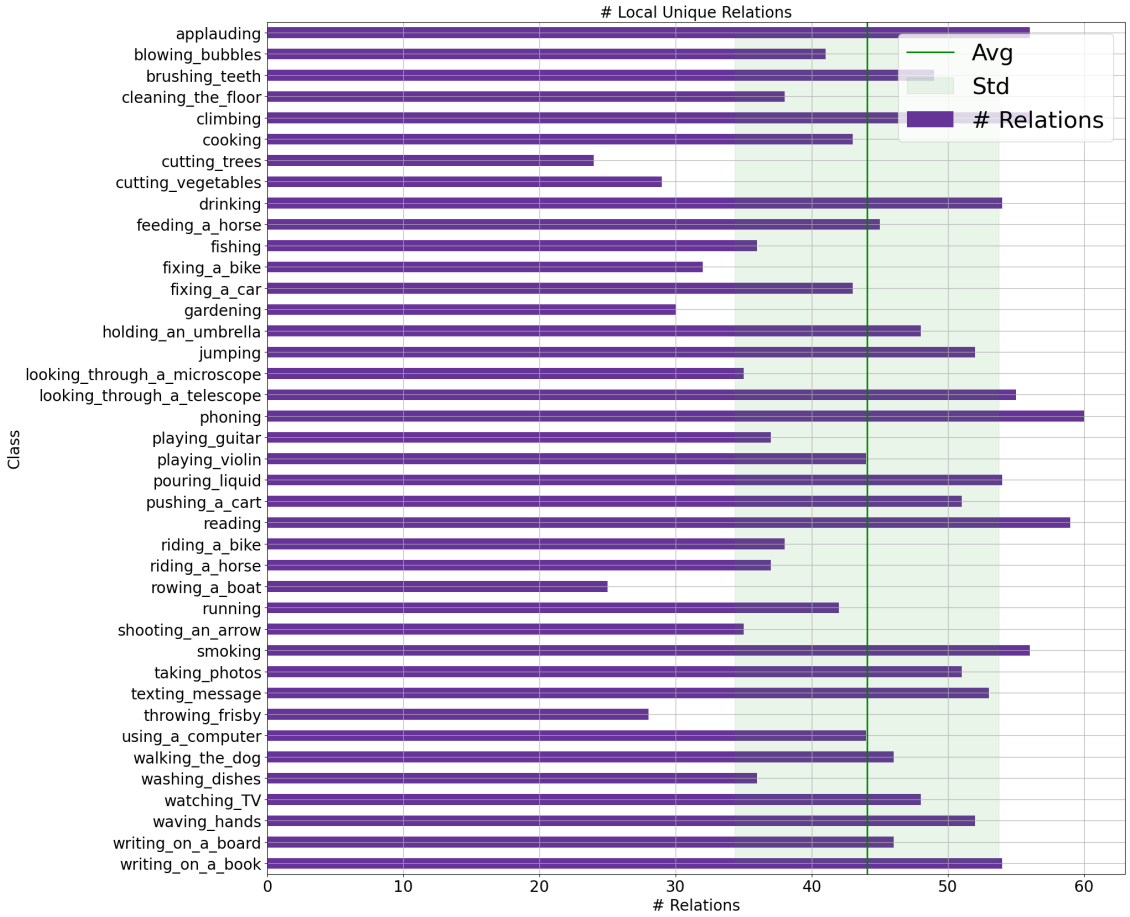

Figure 4.9: Number of Unique Local relations - Action40/ResNet-152

sification task, versus classes with high coverage (such as "riding_a_bike") where significant global relations are present in most of the local instances. Figure 4.12 presents an example of how our method works for the class "riding_a_bike" in the *Action40* dataset and *ResNet-152* architecture, which had the higher coverage.

It is shown that, based on the top concepts extracted from the first component, ten local relations based on an instance of "riding_a_bike" and three global relations for the same label were selected as candidates. In the validation step, images from *Visual Genome*, which contain the concepts presented in the relation candidates, are used to verify which relation was learned by the model. In our example (Figure 4.12), only the "wheel is part of a bicycle" was identified as having been learned. As the relation learned is presented both locally and globally, we then define that the relation covers this case.

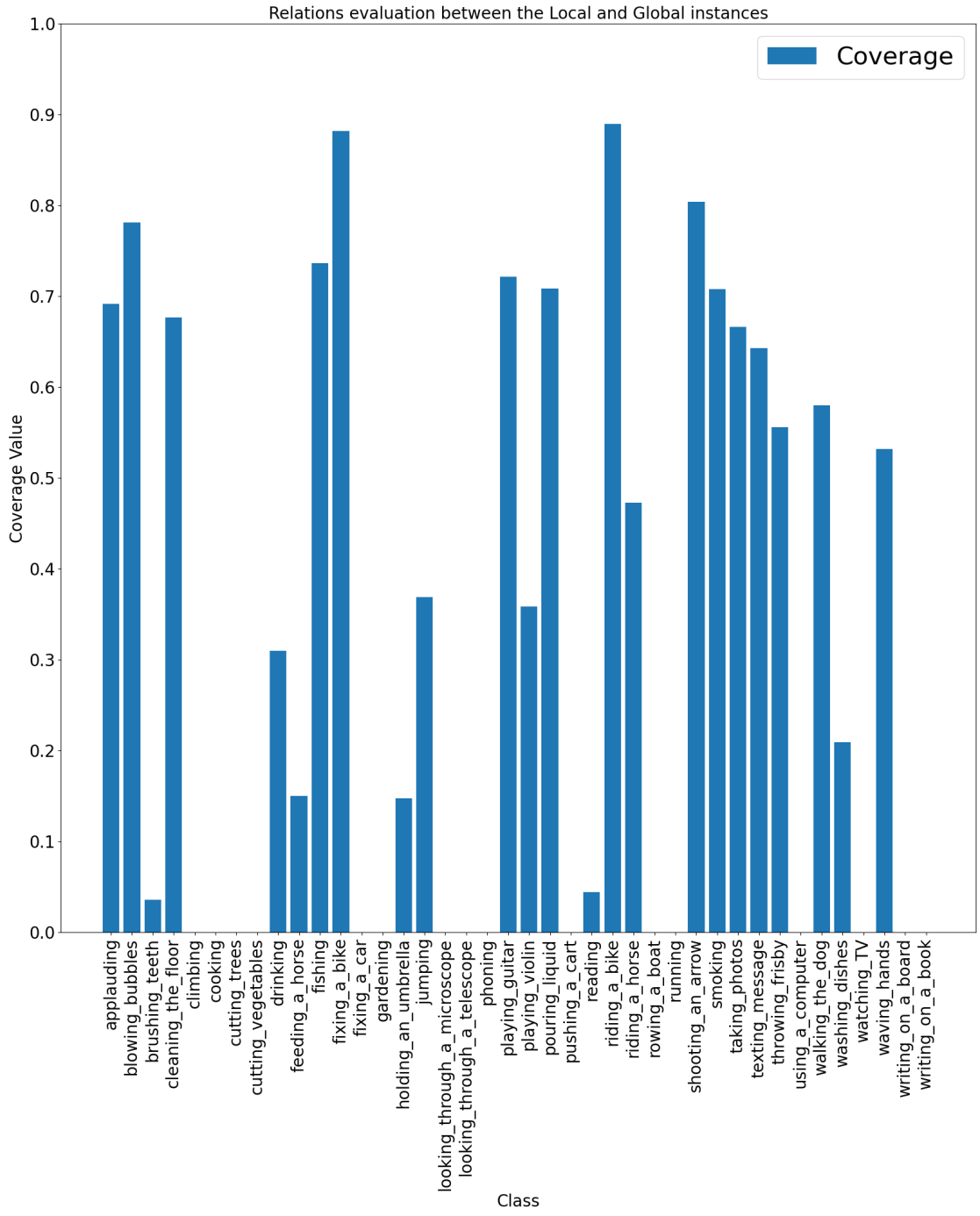

Figure 4.10: Coverage on *Action40*

Discussion about the Coverage. We might be tempted to say that for classes with low coverage, the model is likely to not have learned relations that are crucial for the classification tasks, and this might be used as a starting point to investigate how to better train the model for those classes, for example by looking at class imbalances or data augmentation techniques as well as knowledge injection mechanisms.

One example can be seen when we look for a specific class, such as "fixing_a_car".

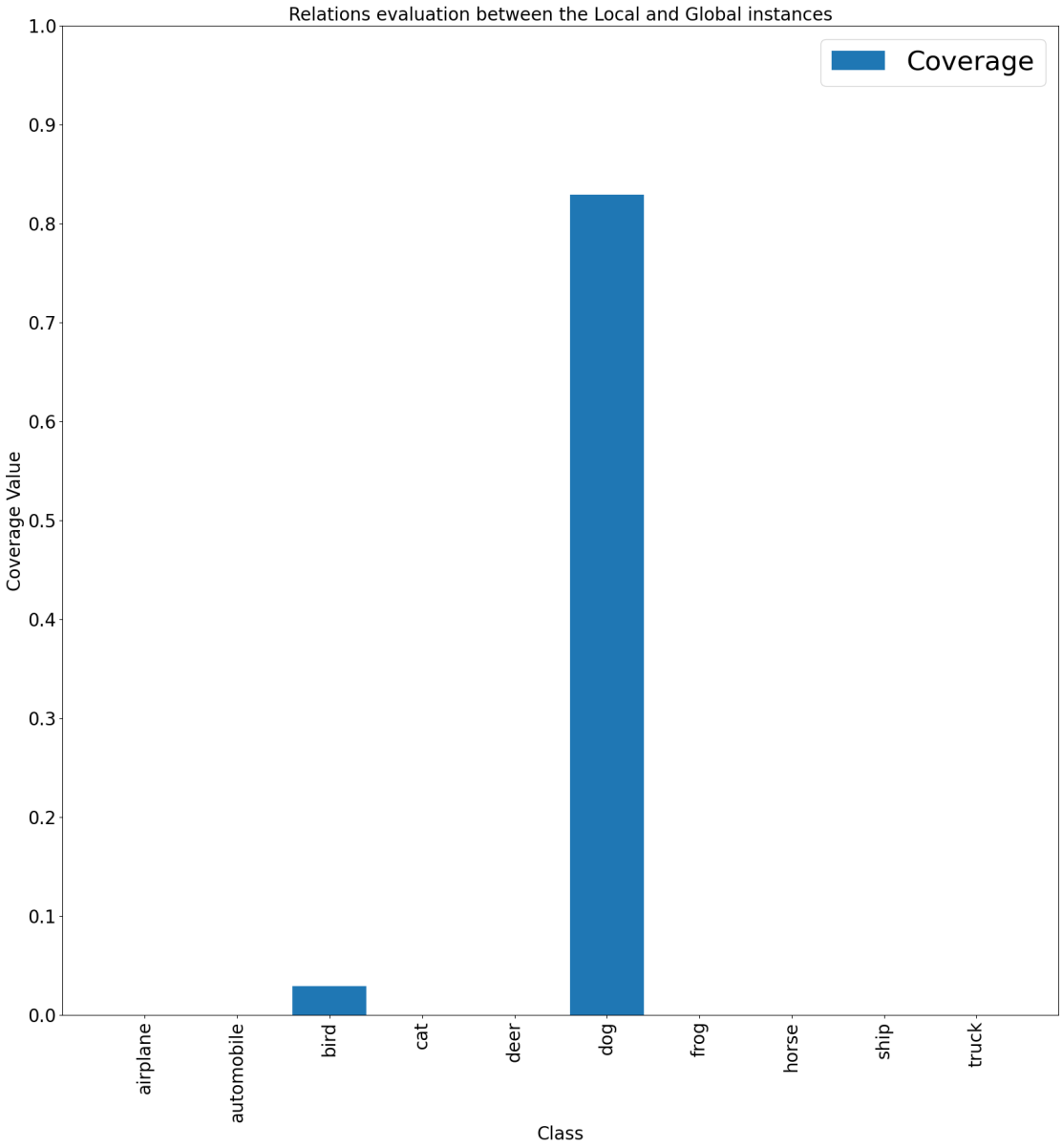

Figure 4.11: Coverage on *CIFAR-10*

There are more than 40 unique local relations, such as "arm is part of a person", which we can see in Figure 4.6, and "wheel is part of a car" (Figure 4.9). However, when we examine the global relations, there are none in Figure 4.8, indicating that none of the global candidates for this class have been validated by the process. When we consider the global candidates for this class, we find the relation "car is related to train", based on the top concepts "car" and "train". This example demonstrates how the concepts "train" and "car" might be regarded as the same, or that the relation ground truth might include some inconsistent samples, where "car" and "train" should have the relation "is a transport". The developer may use

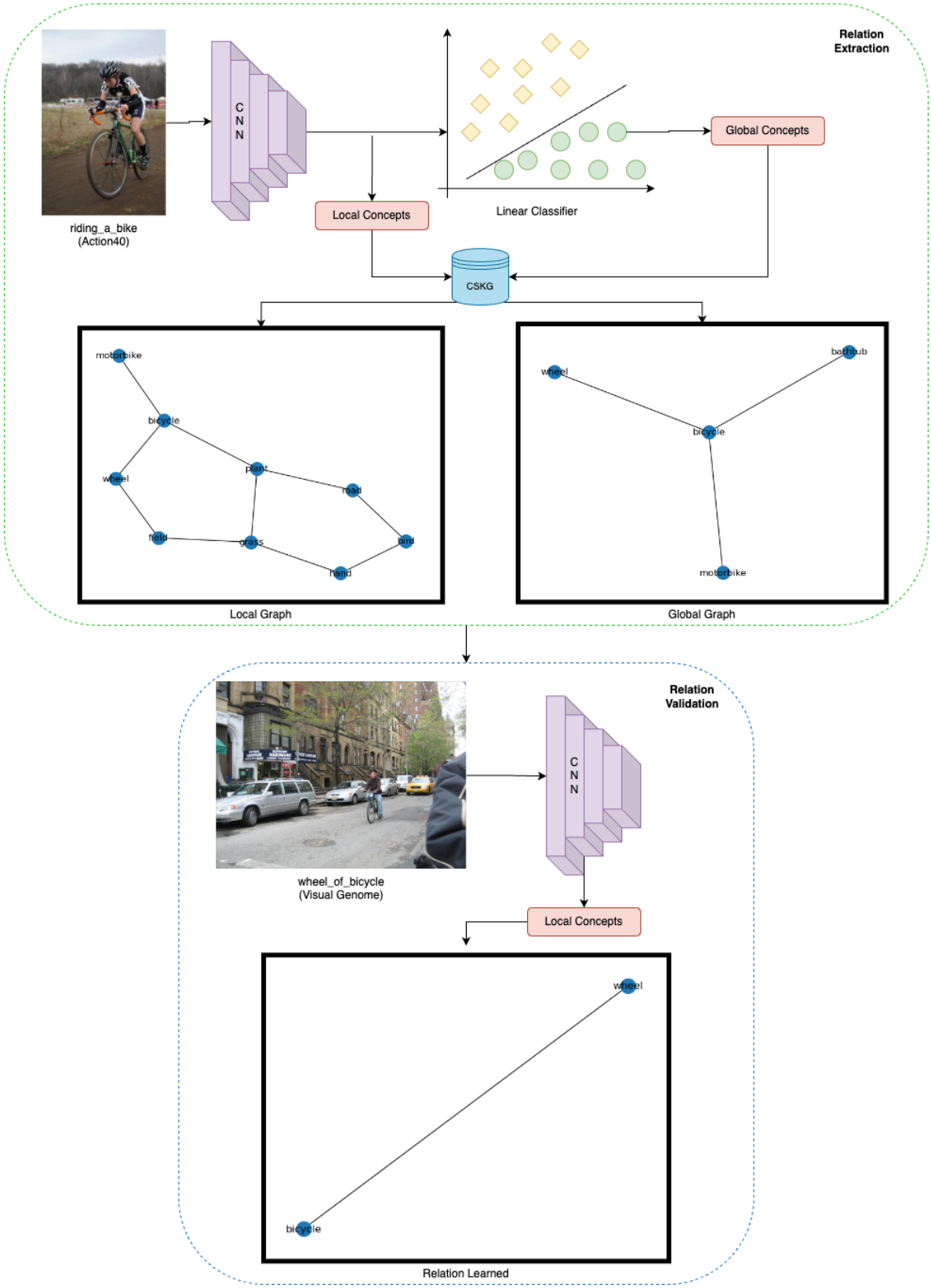

Figure 4.12: Relation Learned Example

this outcome to enhance the model, and a typical user can comprehend the model's judgements.

Another example is the class "watching_TV", which has more than 40 local unique relations identified (Figure 4.9), such as "screen is part of monitor", but which did not uncover any significant relation on global concepts (Figure 4.8). Looking manually, we discovered that the concept "tvmonitor", learned on *Network Dissection*, might be related to the concept "monitor", providing the developer with a pointer as to where they can apply some transformation to solve this problem. Analysing the results of this technique provides transparency to both the trained model and the approach itself, as well as opportunities for improvement. In this case, verifying the concepts' synonyms could provide more relations.

However, we are aware that the values for coverage might also depend on other factors. For example it might depend on how well the linear classifier identifies concepts that separates classes well. It might also be due to the quality of the concept extraction approach based on *Network Dissection*, which in turn might depend on the quality of the dataset. For a low-resolution dataset such as *CIFAR-10*, for example, we observe that only classes "dog" and "bird" have high coverage. This observation would need to be confirmed by conducting an ablation study to identify the quality of concepts and how they affect the overall pipeline, and this is an area for further investigation.

We are aware that a deeper analysis is required to reach this ambitious goal, and we have identified some limitations of our approach that we plan to extend in future work. One limitation of our method is its reliance on *Network Dissection*, which may not fully capture the complexities of human-understandable concepts. This could be addressed by relying on larger datasets with pixel-level concept labels, or by exploring alternative methods for disentangling concepts, such as Concept Activation Vectors [61] and *CLIP-Dissect* [93]. Additionally, there is the need to measure consistency between common-sense knowledge graph relations and the concepts derived from *Network Dissection* and *Visual Genome*, and define additional metrics that

can help identify the suitability of different knowledge graphs.

4.5 Summary

In this Chapter, we started by advocating the importance of relations when it comes to understanding decision processes. In order to have access to such relations, in this Chapter we presented the second component of our framework, which focuses on harnessing the power of external knowledge graphs and multimodal datasets to extract and validate key relations that are influencing *CNN* decision processes. We aim to present a mechanism to identify and single out, across a considerably large set of possible concepts and relations, the few that are relevant to the decision-making process hidden in the *CNN* deep representation.

Hypothesis 2 of this PhD thesis relates to how knowledge in form of a Knowledge Graph (*KG*) can be used to identify the relations learned by a deep model. In order to demonstrate this hypothesis, our methodology involved two main steps, namely i) relations extraction - answering Research Question **(RQ3)** (Section 1.5), and ii) relation evaluation - associated to Research Question **(RQ4)** (Section 1.5).

In the first step, concepts from the model's output are mapped to nodes in a knowledge graph, specifically the Common-sense Knowledge Graph (*CSKG*), to identify direct relations between these concepts. This approach maintained simplicity and relevance by focusing on direct links and avoiding excessive noise. The *CSKG* provided a structured representation of entities and their relations, facilitating the identification of meaningful, common-sense connections between concepts.

In the second step (relation evaluation), we used the *Visual Genome* [65] dataset, which contains images annotated with object and relations, to verify whether the model learned the identified relations. We assessed the model's understanding of these relations by running cropped images through the model and analysing the activation of specific filters corresponding to specific concepts. Our experimental evaluation showed that 12% of the images from *Visual Genome* had at least two identified concepts among the top 10 activated filters from the concept extraction

phase 3.2 for the *Action40* dataset in the *ResNet-152* architecture (Table 4.1), resulting in 2,495 unique local relations and 339 global relations. For the *CIFAR-10* dataset, using the same architecture, our experiment showed that 14% of the images from *Visual Genome* had at least two concepts extracted from the previous component, resulting in 2,176 unique local relations and 139 global relations. Coverage was measured by comparing local and global relations, achieving high coverage for some classes and highlighting areas for potential improvement.

Our experimental results validated our framework's effectiveness and showcased its potential to improve explainability of decision processes in image classification. By integrating common-sense knowledge graphs with *CNNs*, we significantly enhanced the interpretability of image classification models, being able to use relations among concepts that are used in the classification task. This component of the framework, with its promise of promoting model transparency and providing valuable insights for developers, inspires us to refine and advance our models further. Future research should focus on addressing synonym handling and potential biases in concept as well as relation extraction to increase the coverage and reliability of this approach, and using different external knowledge sources aggregated to improve the validation of the relations.

Chapter 5

Neuro-Symbolic Inductive Rule Learning for CNNs Explainability

5.1 Introduction

This Chapter explores the last component of our framework, focused on the extraction of deductive rules that can provide an understanding of the decision process of a *CNN* in a way that is more similar to human reasoning. This component represents a step forward in the generation of explanations about what the model has learned about the task, and how concepts and relations are used to perform such task.

The success and popularity of Deep Learning in Computer Vision for high-risk applications has clearly highlighted the need for transparency and explainability. A focus on human-centred decision-making requires combining powerful data-driven learning with knowledge-driven reasoning, which has recently led to a sudden interest in the field of Neuro-Symbolic Artificial Intelligence (*NeSy*) [40]. *NeSy*, which is also an Association, a Conference Series, and a Journal [1], aims to explore new ways to integrate symbolic logical constructs with neural structures as a game changer for trust, accountability, and sound explainability of modern *AI* solutions. We consider the component described in this chapter to be an instance of Neuro-Symbolic integration, in that we aim to learn logic rules from disentangled concepts extracted

[1] https://www.city-data-science-institute.com/nesy

from deep representations and use these rules as an approximation of the reasoning process of the trained *CNN*.

In order to leverage neuro-symbolic integration to explain deep representations, we need to be able to abstract logic rules and relations from data samples [40]. In what follows we provide a brief introduction to Inductive Logic Programming before delving into our methodology for extracting and validating logical rules from disentangled concepts.

5.2 Background on Inductive Logic Programming (ILP)

Inductive Logic Programming (*ILP*) [27], is a data-driven rule learning approach which learns logical rules that generalise a set of examples [27]. Such rules capture logical inference via generalisation instead of statistical regularities, and therefore learning them requires fewer examples.

An *ILP* task is based on three sets $\{B, E^+, E^-\}$, where the set B comprises the *Background Knowledge*, the set E^+ contains the positive examples and E^- the negative examples (See [27] for a more complete introduction to *ILP*).

If we compare *ILP* to a traditional Machine Learning task on tabular data, we would have that B contains the data features, E^+ contains the positive training targets and E^- the negative training targets. Let's illustrate these concepts with an example taken from [27], and let our tabular dataset be the one illustrated in Table 5.1. For this dataset, let's consider "job" and "enjoys lego" as features and let "happy" be the target variable (or label) we want to predict. In *ILP*, data is not represented as table but as a set of logical rules (a logic program). Our *Background Knowledge* B for the example is similar to features but can represent properties related to the examples.

We can represent the data in Table 5.1 as the background knowledge set B:

Name	Job	Enjoys Lego	Happy
alice	lego builder	yes	yes
bob	lego builder	no	no
dave	estate agent	no	no

Table 5.1: Dataset Example [27]

$$B = \begin{pmatrix} lego\_builder(alice). \\ lego\_builder(bob). \\ estate\_agent(dave). \\ enjoys\_lego(alice). \end{pmatrix}$$

In this case, the set of positive E^+ and negative E^- examples with respect to our target label can be represented as:

$$E^+ = \begin{pmatrix} happy(alice). \end{pmatrix}$$

$$E^- = \begin{pmatrix} happy(bob). \\ happy(dave). \end{pmatrix}$$

Given these sets, the *ILP* goal is to induce a hypothesis that entails as many positive examples and as few negative examples as possible. The *ILP* builds hypotheses by generalising from specific positive and negative examples using background knowledge. For example, one logical rule generated can be represented as:

$$H = \{\forall A.\ lego\_builder(A) \wedge enjoys\_lego(A) \rightarrow happy(A)\}$$

This indicates that all instances that are *lego_builder* and *enjoys_lego* are considered *happy*. This is an example of how a hypothesis is generated based on the background knowledge and examples. The next step will be to ensure the hypothesis covers all the positive examples and none of the negative ones. In this example, this hypothesis

achieves the expected entailment, and we can deduce this logic rule as the knowledge learned by the program. The method described here is known for learning from entailment settings, where the *ILP* system imposes stringent restrictions: the rule must cover all positive examples in the learning process.

When the set of examples is small, the rule learning process with *ILP* can find hypotheses that represent the data very well; however, as the number of examples grows or becomes more complex, the rule learning process becomes more error-prone and the quality of the learned rules degrades. In order to mitigate this problem, approaches have been proposed that combine *ILP* with more robust and expressive declarative paradigms, such as Answer Set Programming [25]. In this work, we focus on this particular combination where *ILP* aims to learn Answer Set Programs (*ASP*), which support non-monotonic and common-sense reasoning. For an introduction to *ASP* please see the work presented in [36].

Inductive Learning of Answer Set Programs

Answer Set Programming (*ASP*) [76] is a declarative programming approach that uses stable model semantics [42] and is designed for *NP-hard* search problems. Within the scope of this thesis, answer set programs consist of a set of rules of the following form:

$$h \text{ :- } b_1, b_2, ..., b_n, not\ c_1, not\ c_2, not\ c_m. \tag{5.1}$$

where $h, b_1, ..., b_n, c_1, c_m, m \geq 0, n \geq 0$ are atoms and $h \notin \{b_1, ..., b_n, c_1, ...c_m\}$.

With this restriction, answer set programs accept only one answer set computed with the forward chaining procedure. The procedure initialises the answer set X with trivially satisfied rules (of the form h.). Then, the answer set is updated with heads of the satisfied rules. Note that a rule is satisfied if every $b_i \in X$ and no $c_i \in X$. The procedure repeats until no more rules can be satisfied.

FastLAS [71] is a state-of-the-art system for learning answer set programs capa-

ble of solving *Observational predicate learning* (*OPL*) tasks over large search spaces. *OPL* tasks aim to learn concepts that are directly observable from given examples, and their solutions are *ASP* programs, called *hypotheses*, that define such concepts. The approach in *FastLAS* and its extensions has been applied to learning rules from deep neural networks in various domains, including material discovery [30], reasoning tasks with images [29], and cyber-security [34].

The Hypothesis Space. A common method used in *ILP* systems to reduce the size of the hypotheses set, while focusing on meaningful rules that follow some expected pattern, is the use of mode declarations as language bias to define the search space (also called *hypothesis space*) of a learning task. A mode bias M specifies two sets, M_h and M_b, of a mode head and body declarations, respectively. Informally, a mode declaration is an atom whose arguments are either `var(t)` or `const(t)`, where `t` is a constant called a *type*. An atom is compatible with a mode declaration m if it uses the predicate of m and replaces every argument `var(t)` in m with a variable of type `t`, and every `const(t)` in m with a constant of type `t`. Body declarations can be negated atoms using negation as failure. *FastLAS* also uses mode declaration to define the hypothesis space.

Definition 1 (Hypothesis space) *Let $M = \langle M_h, M_b \rangle$ be a mode bias. The hypothesis space $S_M = \{r_i | i \geq 1\}$ is the set of rules r_i such that head(r_i) is compatible with a mode head declaration, and each literal in body(r_i) is compatible with a mode body declaration.*

The mode declaration is defined in the background knowledge, and an example can be described as follows:

$$modeh(predicate)$$
$$modeb(2, predicate)$$

where *modeh* is M_h and defines that a rule should have a specific *predicate* as its head, and *modeb*, which represents M_b, defines that a rule should have in its body the *predicate* with two arguments.

Examples as Partial Interpretations. Another possible method for the learning process using *ILP* system is learning from interpretations [27]. In this setting, examples are defined as a set of facts — interpretations — which can be complete or partial. Complete interpretations provide all relevant facts about the entities involved, while partial interpretations provide only a subset of the facts. A *partial interpretation* e is a pair of sets of ground atoms $\langle e^{inc}, e^{exc} \rangle$, where e^{inc} represents the positive examples and e^{exc} the negative ones. A interpretation (i.e. a set of ground atoms) I can *extends* e iff $e^{inc} \subseteq I$ and $e^{exc} \cap I = \emptyset$, ensuring logical consistency defined by the inclusions and exclusion.

For example, if we are trying to learn rules about animals, a complete interpretation might specify everything about an animal, such as whether it has feathers, can fly, and lays eggs, among others. On the other hand, a partial interpretation presents only specific facts about an animal. This method enables one to learn without much data or background knowledge, allowing one to work with an incomplete dataset.

The partial interpretations can also be defined based on the context in which they belong. This method is called *weighted context-dependent partial interpretations* (*WCDPIs*) [70], which are weighted according to the context. A *WCDPI* can be represented as a tuple $e = \langle e_{id}, e_{pen}, e_{pi}, e_{ctx} \rangle$, where e_{id} is an identifier for e, e_{pen} is either a positive integer or ∞, called a penalty, e_{pi} is a partial interpretation, and e_{ctx} is an *ASP* program called a *context*. A *WCDPI* e is *accepted* as an answer set by a program P if there is an answer set of $P \cup e_{ctx}$ that extends e_{pi}. In *OPL* tasks, the predicates in the examples' e^{inc} coincide with the predicates defined by the learned hypothesis, and programs accept exactly one answer set. The acceptance also prioritises interpretations during the learning process based on the penalties.

The Learning Task. *FastLAS* is a noise-tolerant Inductive Rule Learning system that learns an *ASP* program H, called a *hypothesis*, from *WCDPI* examples. If a hypothesis H does not accept a *WCDPI* example, we say that it *pays the penalty* for that example. Informally, penalties are used to calculate the cost associated with

a hypothesis for not covering examples. The cost function of a hypothesis H is the sum of the penalties for all examples that are not *covered* by H, augmented by the length of the hypothesis.

A *FastLAS* learning task with noisy examples - which means examples that can be mistakenly labelled or are ambiguous in a specific domain - consists of an *ASP* program composed by a i) background knowledge, ii) a hypothesis space defined by a language bias (*modeh* and *modeb*), expressing the set of rules that can be used to construct a solution to the task, and iii) a set of *WCDPI* examples. Such a task aims to find a hypothesis H in the hypothesis space that minimises a cost function concerning a given set of noisy examples. The hypotheses should be mutually exclusive and are tested one by one in order to verify how many examples each hypothesis can cover. The final result is the set of hypotheses that satisfies the maximum number of examples with the minimal penalty. This is formally defined below, adapted from [71].

Definition 2 (LAS task) *An* Observational Predicate Learning from Answer Sets *task is a tuple* $T = \langle B, S_M, E^+ \rangle$*, where* B *is a Background Knowledge (which can also be an ASP program - AS),* S_M *is a hypothesis space, and* E^+ *is a set of WCDPI such that* $\forall e \in E^+$, $|AS(B \cup e_{ctx})| = 1$*, and no predicate in* M_h *occurs in* M_b *or in the body of any rule in* T*, avoiding cyclic definition. Given a hypothesis* $H \subseteq S_M$*,*

1. $\text{UNCOV}(H, T)$ *is the set consisting of all examples* $e \in E^+$ *such that* $B \cup H$ *does not accept* e*, meaning the hypotheses that do not cover all positive examples.*
2. *The penalty of* H*, denoted as* $\text{PEN}(H, T)$*, is the sum* $\sum_{e \in \text{UNCOV}(H,T)} e_{\text{pen}}$.
3. *The score of* H*, denoted as* $\mathcal{S}(H, T)$*, is calculated as* $|H| + \text{PEN}(H, T)$.
4. H *is an* optimal inductive solution *of* T *if and only if there is no* $H' \subseteq S_M$ *such that* $\mathcal{S}(H', T) < \mathcal{S}(H, T)$.

The optimisation function used by *FastLAS* aims to learn a hypothesis H that jointly minimises the total penalty paid for the uncovered examples and hypothesis

length ($|H|$). In practice, this creates a bias towards shorter and, therefore, more general solutions that cover examples with a high penalty value, featuring shorter rules capable of considering uncovered examples.

Now that we have introduced *FastLAS* and the *ILP* system used, we are going to discuss how these elements are employed in our approach to extract and validate rules from disentangled concepts. To achieve this, we propose a new component that uses the model's concepts and relations through disentanglement, with rule learning via *ILP*.

The main contributions presented in this chapter are: (i) a neuro-symbolic approach to extract deduction rules from a trained *CNN* model, using concepts and relations from the previous steps of the framework **(RQ5)** (Section 1.5); and (ii) an evaluation aiming to verify if the rules extracted can be used to understand the inner workings of the model **(RQ6)** (Section 1.5). The novelty of this process lies in the integration of concept and relation extraction with a state-of-the-art *ILP* approach to map the knowledge acquired by the model in form of rules. In addition, for this step we do not rely on network dissection for concept and relation extraction, and used a state-of-the-art visual-language model, showing the flexibility of our approach in combining different methods for the different phases. The implementation of the component is available online[2].

5.3 Extracting Rules (RQ5)

This Section presents our approach to extracting rules regarding what an image classification model has learned. In addition, to demonstrate the framework's coverage, we also use a different method for extracting the concepts [94] and relations starting from the same architectures (*ResNet-152* and *ResNet-50*) and dataset (*Action40* and *CIFAR-10*) previously used in this research (Chapter 3). The extraction method created in [94] does not rely on a static concept dataset, as in *Network Dissection*, and it was used to create the input to the rule extraction step, whereupon

[2]https://github.com/EricFerreiraS/SEVC

both - concept and relation extraction - are performed at the same time. This method leverages Large Language Models *(LLMs)*, which are advanced language models with many parameters and exceptional capabilities [21], able to predict text based on a context given by a prompt. Furthermore, this alternative method aligns with those described in the components proposed earlier in this work, extracting high-level features learned by the pre-trained model.

In the process of extracting logical deduction rules that describe the trained *CNN's* decision-making process, the method introduces an additional step when extracting concepts [94]. This step starts with classes and interrogates a Large Language Model using specific prompts. As a result, we obtain a rich set of concepts and relations that do not require extensive human labelling. Starting from this set, we use a text-image model to determine how the concepts and relations extracted from the *LLM* are relevant to the trained *CNN* in the classification task. It is only the most relevant concepts and relations that are then used for inductive rule learning. The details of each step in this process are provided in the remainder of this section.

***LLM* prompting for candidate concepts and relations.** Consider a dataset $\mathcal{D} = \{x_i | 1 \leq i \leq N\}$ where each image $x_i \in \mathcal{D}$ has a class label $l_i \in L$. For the concept and relation extraction step, we provide the set of class labels to the *GPT-4 LLM* [1], which returns, for each label, the concepts and relations relevant to the class, as in [94, 28]. For each class label $l_i \in L$ we use the following prompts:

- List the most important features for recognising something as l_i
- List the things most commonly seen around l_i
- Give super-classes for the word l_i

The output is a concept and relation set $\mathcal{T} = \{t_1, ..., t_M\}$ containing all the concepts and relations for label l_i.

Text-image alignment of concepts and relations with CLIP-Dissect. The outcome obtained by using all three prompts above is provided as input to the *CLIP-Dissect* approach, a method that also produces disentangled representations, similar to the *Network Dissection*, but it uses a text-image model to compute a concept/relation matrix [93], which represents the concept/relation learned by each filter. *CLIP-Dissect* uses an image encoder Enc_I and a text encoder Enc_T from a *CLIP* [98] model to generate a matrix that results from their inner product. The concept/relation matrix derived from the inner product, which provides the similarity between the image and the text, is passed through a specific *CNN* model that was previously trained for a given classification task, and any similarity function can be used to associate a neuron to a concept/relation.

Each *CNN* layer is composed of a set of neurons $\mathcal{K}$, and the number of neurons varies depending on the model architecture. An activation map $A_k(x_i)$ is a matrix that represents the neuron's activation for a specific input x_i and a given neuron $k \in \mathcal{K}$. Let us consider a mean summary function g, which computes the mean of the values in the activation maps $A_k(x_i)$ and returns a real number, representing the neuron activation as $g(A_k(x_i))$.

An activation vector for a neuron k is defined by the result of the function g for all inputs N, and is represented as $q_k = [g(A_k(x_1)), ..., g(A_k(x_N))]^\top, q_k \in \mathbb{R}^N$. The similarity function calculates how alike a concept/relation t_M is to an activation vector of a target *CNN's* neuron k, based on the highest value for each pair of concept/relation and neuron. As the previous concept and relation extraction's steps (Chapters 3 and 4), we use the same *CNN* where the concepts and the relations are associated with the neurons, and for each image x_i in the dataset, we pass it through the network and extract its activation vector $v_i = [g(A_1(x_i)), ..., g(A_k(x_i))]$, where k depend on the *CNN's* number of neurons and $i \in \mathbb{R}^N$.

For each v_i, we rank the highest values by index, assuming these are the most relevant neurons (hence the most important concepts/relations) for x_i. We now have a list of concepts/relations that have been learned for each given image x_i, along

with a similarity score indicating how well these concepts/relations were learned. These are the concepts/relations we can use as positive and negative examples for the inductive rule learning process.

Rule Learning with FastLAS. We make use of *FastLAS* [71], an *ILP* system for rule learning. As mentioned in Section 5.2, a *FastLAS* task consists of background knowledge, language bias and a set of positive examples. To encode concepts/relations extracted from the *CNN*, we use the following predicates:

1. `label/1` - Predicate representing a class. For example, `label(riding_a_bike)` represents the class "riding_a_bike" in the *Action40* [131] dataset.

2. `concept-relation_id/1` - Assigns ids to concepts/relations identified in extractions phase. For instance, `concept-relation_id(2)` represents the relation "arm part of a person" in the *Action40* dataset.

3. `selected/1` - Class believed to be true for a specific data point. For example, in *Action40* dataset, one possibility would be any of the 40 classes.

The *background knowledge* is used to provide relevant pre-existing knowledge to the learner. It is kept minimal in this scenario, consisting of the definitions for `label/1` and `concept-relation_id/1`, which are used to constrain the search space. In addition, we also define that only one class should be selected, to prevent a single rule from representing multiple classes (and this constraint is also part of the background knowledge):

```
:- selected(X), selected(Y), X != Y.
```

The *language bias* specifies that any rule with `selected(class)` as a head and 0 or more `concept-relation_id(k)` in the body can be learned. The declarations are presented below, respectively.

```
#modeh(selected(const(label))).
#modeb(concept(const(concept-relation_id))).
```

We generate one *FastLAS Weighted Context-Dependant Partial Interpretation (WCDPI)* example per data point, where each Partial Interpretation represents the set of concepts/relations present in each image. Remember that in the Text-image alignment of concepts and relations with *CLIP-Dissect* step (Section 5.3), the t concepts/relations and a list containing their similarities (i.e. level of confidence of the neuron capturing a certain notion) are returned. The extracted concepts and relation (i.e. their IDs) form the context of a *WCDPI*. Similarly to [29], we aggregate the similarities of the concepts/relations included in the context of an example, using the minimal similarity value between all the concepts/relations within the context. The result of this aggregation defines the penalty of the *WCDPI*. The inclusion of a *WCDPI* example contains the `label` predicate with the true class. The exclusion predicate contains `label` predicate with all incorrect classes for the data point.

For instance, we can represent a *WCPI* as follows:

```
#pos(ex_2494_riding_a_bike_266@1,
{ selected(riding_a_bike) },
{ selected(taking_photos),  selected(playing_violin),
selected(climbing) ...},
{
concept-relation(44).
concept-relation(61).
concept-relation(66).
concept-relation(74).
concept-relation(199).
concept-relation(200).
concept-relation(232).
concept-relation(273).
concept-relation(363).
}).
```

where we have a positive example for the image "riding_a_bike_266" in the *Action40*

dataset, with $e_{id} = 2494$ (id) and $e_{pen} = 1$ (penalty). The e_{pi} is represented by the inclusion $e_{inc} = selected(riding\_a\_bike)$, which denotes the class to which the image belongs, and $e_{exc} = \{selected(taking\_photos), ...\}$ for the other classes not represented by the image. Finally, the $e_{ctx} = \{concept - relation(44), ...\}$ represents the set of facts - encoded as integers - that depict the image.

Given the background knowledge, the language bias and the set of examples, running *FastLAS* [71] produces a set of rules. A possible rule returned by *FastLAS* may be:

```
selected(riding_a_bike) :- concept-relation_id(74), concept-relation_id(59).
```

where *ids* 59 and 74 represent "body strength" and "gloves", respectively.

Given a data point at test time, we generate `concept-relation_id` atoms in the same manner. The generated atoms are passed to the answer set solver (e.g. *clingo* [41]) together with the learned rules. The predicted class is contained within the `selected` predicate included in the answer sets of the background knowledge extended with the learned rules. Note that this approach may produce zero classes when the solver cannot find a stable model, or when the problem is unsatisfiable; or it may yield more predicted classes when the solver finds multiple models for the solution. The evaluation step proposes a way to deal with it partially. However, there are other options, such as learning probabilistic rules, to overcome such outcome.

At the end of the process, a set of rules that the rule extraction step has found to be the best for covering the examples used is generated, followed by the examples that are not covered by the solver.

5.4 Rule Evaluation (RQ6)

In this Section, we present the experiments executed to evaluate the last component of our framework, which is responsible for extracting deductive rules that approximate what the deep model has learned in a declarative logic way. This aims to validate our answers to **(RQ5)** (Section 1.5) and **(RQ6)** (Section 1.5).

The ultimate goal of the rule extraction component is to be able to generalise well so that it can emulate the behaviour of the last layer of the trained *CNN* by performing the classification task through deductive inference, as this would be explainable by design. There are two ways to assess the rule-learning process: (i) similar to another classification task, measures such as accuracy and precision can be utilised, concentrating on learner performance; (ii) the second approach focuses on the rules generated by the process, evaluating both their length and the total number of rules [16].

In this work, we focused on the effectiveness of the rule learner's generalisation, evaluating the method's accuracy by assessing how many rules correctly identify the class based on the provided concepts and relations as input. Given the possibility of producing zero or more output classes, we extended the regular accuracy to the top-5 accuracy, whereby we consider a result to be correct if it is present in the top five classification outputs. Another relevant aspect to evaluate is the number of examples used in the rule-learning's training stage.

This is a crucial aspect, as the deep model is known to be data-hungry, requiring a high volume of training data to achieve good accuracy. Conversely, rule-based approaches that generalise well need far fewer data samples. Hence, if we can learn rules from a small number of (positive and negative) examples and achieve good classification accuracy with those rules, we could address the data scarcity problem.

To take the sample size aspect into account, we compare the accuracy obtained by our model across training samples of varying sizes, inspired in different works [114, 38, 111]. This allows us to verify the generalisation capability of the approach, as better performance with fewer samples means that the model generalises well.

5.4.1 Datasets

The datasets used in the evaluation include *Action40* [131], used in previous Chapters, as well as *CIFAR10* [67]. While *Action40* is not commonly used as benchmark given the complexity of the dataset and the number of classes and samples, *CIFAR10*

is a widely used benchmark dataset for image classification tasks [93, 94].

Using *GPT-4* [1] as *LLM* model, the number of unique concepts/relations created for *CIFAR-10* was 175, while for *Action40* was 661. An example of the extracted text using *LLM* is the notion of "a person with a cigarette in their hand", which contains more than one concept as well as a relation between two concepts.

This is an example of the output from the framework's concept and relation extraction component, but using a different method from those presented in Chapters 3 and 4. Instead of using *Network Dissection* for concept extraction and *Knowledge Graph* for the relation step, we used a combination of a multi-modal approach (*CLIP*) with a large language model. The different approaches used for concept and relation extraction reinforce the main point of the framework, which aims to extract semantic representations based on the concepts and relations that the model has learned.

5.4.2 Setup

For the concept and relation extraction phases, we used the *CLIP RN50* [93] model to encode images and text representing the concepts and relations generated by the *LLM* model. The *CNN* architecture used was *Resnet-152* [51] pre-trained on *ImageNet* [31]. We chose to use the same pre-trained model as the other related works [94, 37] and Chapters 3 and 4 to guarantee a fair comparison and also to speed up the *CNN* learning process. As discussed previously in Section 2.1.1, when the bottleneck layer is used in the concept/relation extraction, we may lose some concepts, given the bottleneck process that adds a new layer at the end to converge the outputs into a compressed and controlled layer; hence we removed the bottleneck layer from the process. We also adopted the Soft Weighted Pointwise Mutual Information *(SoftWPMI)* as a similarity function to associate concepts/relations to neurons based on the *CLIP* encoding [93].

Similarly to [37] and as discussed in Chapter 3, we selected the top 10 most important concepts/relations based on the activation vector v_i resulting from the

extraction phase and used these concepts when generating the positive examples for the rule learning phase. As already specified earlier in this section, we experimented using different sample sizes, in order to measure the variations in performance for the learning process according to different dataset sizes. The samples were uniformly distributed for each class, creating a balanced dataset for the symbolic learning.

For the rule learning process with *FastLAS*, we set the ratio between the scales of the examples and rule penalties to 50, representing the emphasis that the learner will place on ensuring all rules are true. This ratio constitutes a scalability trade-off, as a lower value may result in a higher round error in the penalty calculation, which could require more time to complete. We also set up a time window of less than 72 hours, restricting the output based on the rules learned during this period. We defined this time window based on the size and complexity of our data, as a small dataset might require up to 30 minutes to complete the rule learning process [71]. We also conducted tests for shorter and longer periods, which did not lead to finding better solutions.

5.4.3 Results and Discussion

We evaluate the accuracy of our approach over a test set comprising 10,000 images in *CIFAR-10* and 5,532 images in *Action40*. Figures 5.2 and 5.3 demonstrate the accuracy of each dataset and approach on the two datasets respectively. As in the experiment using the *Action40* it is difficult to analyse the differences between the approaches, Tables 5.2 and 5.3 show the values for a better analysis. To guarantee a fair comparison, we run all our experiments for all the variants considered on both *Resnet-50* and *Resnet-152*.

The *Standard* and the *Label-Free CBM* setups are as presented in [94]: the former uses a sparse layer after the concept/relation extraction, while the latter implements a concept/relation bottleneck layer after the concept/relation extraction and before a sparse classification layer. The *DRCR*, which stands for the Disentangled Representation to Concept Ranking approach, relies on a linear *SVM* classification layer

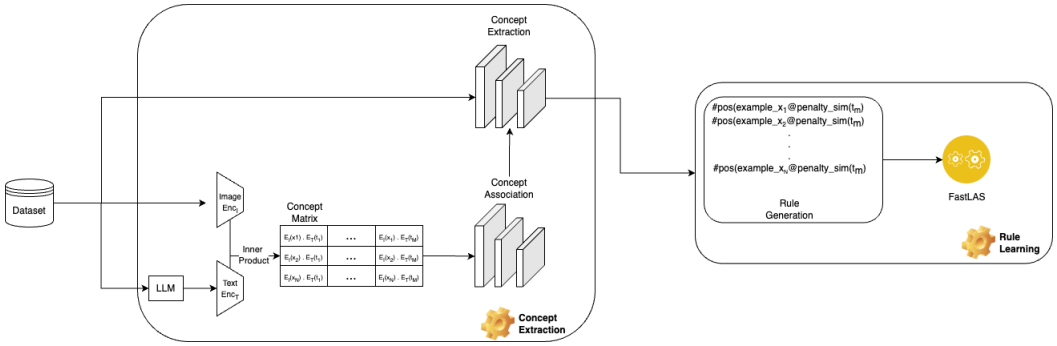

Figure 5.1: Rule Learning process developed in this Chapter

after the concept extraction/relation phase [37], which is also the approach used in Chapters 3 and 4 of this thesis (Figure 3.2). Figure 5.1 illustrates the different approach used in this experiment, where the dataset contains the images and labels, used to generate the concepts/relations associated to the *CNNs* neurons and for the concept/relation extraction from the model. In the results, we will refer to this approach as *NeSy-IRL* (Neuro-Symbolic Inductive Rule Learning).

	Samples				
	100	500	1000	2000	4000
NeSy-IRL	0.0584	0.1316	0.1606	0.1981	0.2060
Standard (sparse)- CLIP_RN50	0.1202	0.1000	0.1000	0.1000	0.1000
Standard (sparse)- ResNet152	0.1000	0.1544	0.1000	0.1000	0.5843
Label-Free CBM - CLIP_RN50	0.1959	0.4692	0.5246	0.5225	0.5842
Label-Free CBM - ResNet152	0.4264	**0.8848**	**0.8869**	**0.8969**	**0.8968**
DRCR	**0.5489**	0.6427	0.6591	0.6717	0.6753

Table 5.2: Accuracy for the CIFAR-10 dataset

In Figure 5.2, we can see how the *Label-Free CBM* with a deeper backbone (*Resnet-152*) performed very well on the *CIFAR-10* dataset, only dropping when trained with less than 500 samples. We noticed that the other methods improved as we added more samples. An exception to this trend can be observed looking at the use of a sparse layer on the *CLIP_RN50* model as in [94]. This behaviour may occur either because the model is overfitted, meaning it can only be improved with much more data, or it has reached its maximum generalisation. The other factor

	Samples				
	100	500	1000	2000	4000
NeSy-IRL	0.0190	0.0278	0.0372	0.0491	0.0929
Standard (sparse)- CLIP_RN50	0.0333	0.0333	0.0441	0.0443	0.0419
Standard (sparse)- ResNet152	0.0333	0.0333	0.0448	0.0490	0.0508
Label-Free CBM - CLIP_RN50	0.0336	0.0486	0.0486	0.0504	0.0542
Label-Free CBM - ResNet152	0.0383	0.0468	0.0325	0.0488	0.0553
DRCR	**0.4248**	**0.6247**	**0.6706**	**0.6923**	**0.7178**

Table 5.3: Accuracy for the Action40 dataset

may relate to its nature, as reducing the number of connections and parameters could cause it to lose the ability to capture complex representations [77].

Figure 5.3 shows how performance is relatively low for all approaches on *Action40*, except when a linear *SVM* is employed for the classification task in the *DRCR* approach. In Table 5.3, we can observe that the values for the approaches, apart from *DRCR*, are very similar to one another. The dependency on a huge amount of data to train a *CNN* is responsible for such low accuracy results for almost all the approaches presented in Table 5.3, while the *NeSy-IRL* approach suffered due to the quality of the concepts/relations extracted, as it could not find rules that generalise the classes effectively. Additionally, the *DRCR's* results can be explained by the fact that the linear separation provided by the *SVM* approach can manage noise or outliers in a small number of data points more effectively than the other solutions, while the *NeSy-IRL* approach shows slight improvement compared to the other approaches when the number of samples increases to 4000, although it still achieves fairly low accuracy.

Our experimental analysis reveals how state-of-the-art approaches achieve good accuracy on a dataset with fewer classes and a huge amount of data - which can decrease outliers and noise - such as *CIFAR10*, despite the inability of these methods to capture logical dependency between the concepts/relations. When the classes in the dataset increase and we need more training data, methods using a sparse layer

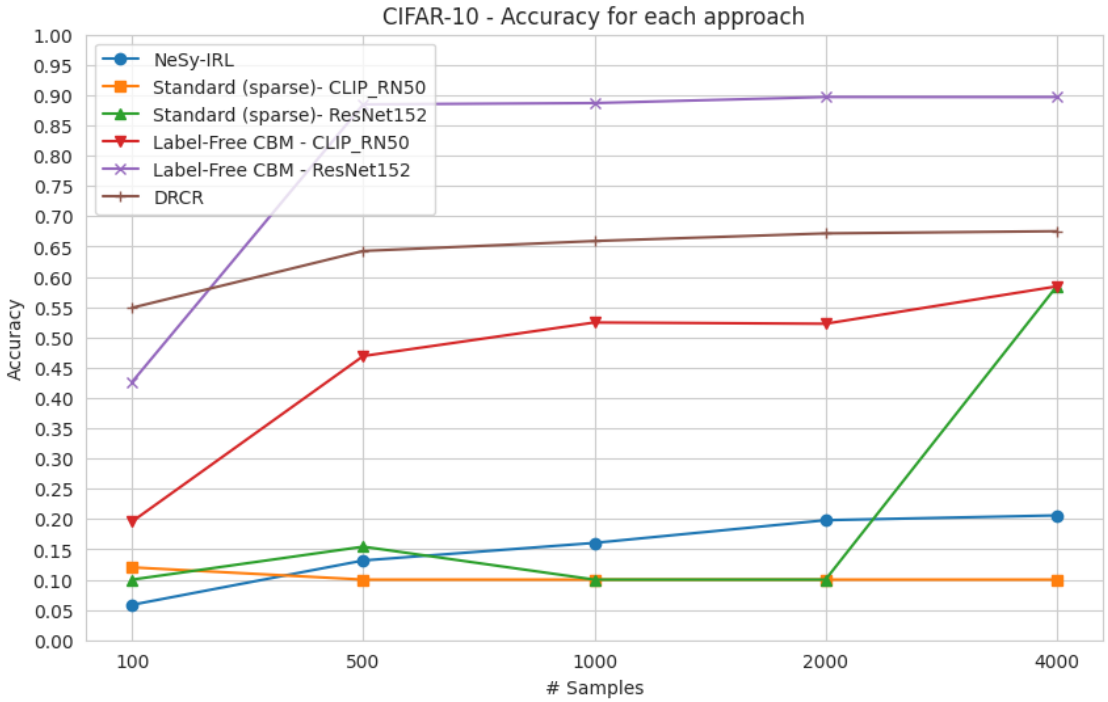

Figure 5.2: Accuracy on the CIFAR10 dataset

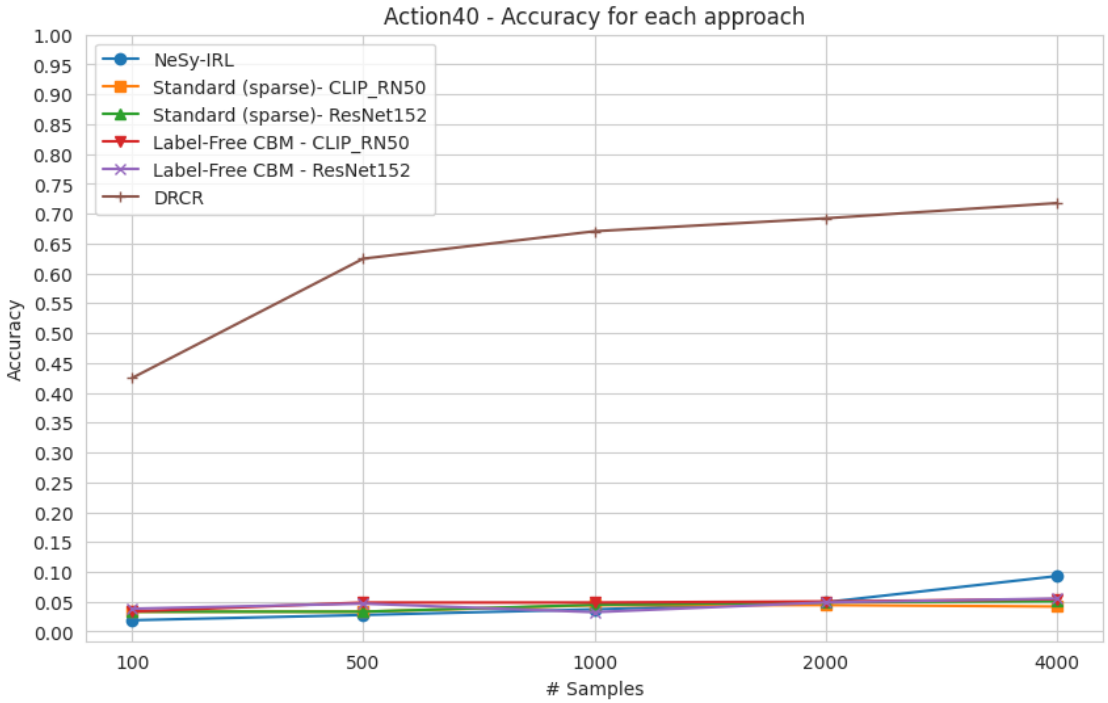

Figure 5.3: Accuracy on the Action40 dataset

(with or without a bottleneck) also fail miserably in terms of accuracy.

Using *FastLAS* as a rule learning method for classification should overcome these two issues since it does not require additional data to learn and generates rules that could capture concept dependency. However, we could not achieve a satisfactory

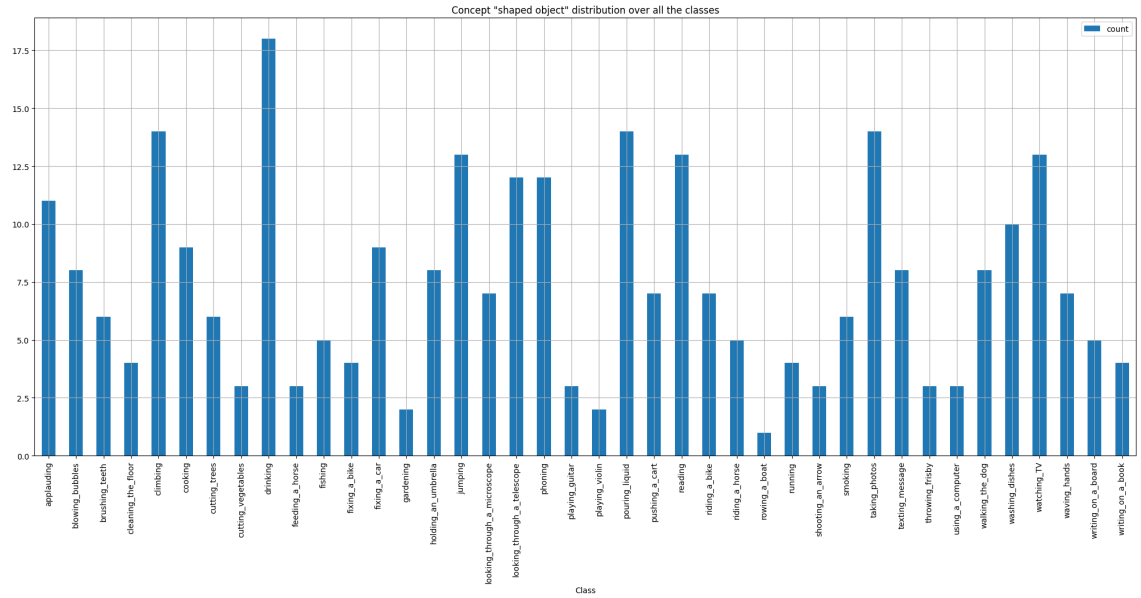

Figure 5.4: Concept/Relation distribution per class - Action40 dataset

result in our experimental evaluation.

One reason for such an outcome can be attributed to multiple overlaps between concepts/relations across classes. Due to this phenomenon, the concepts/relations extracted from the *CNN* are not representative enough for the rule learning process to be able to distinguish between classes. If we consider, for example, the concept/relation "shaped object", this appears in almost every class of the *Action40* dataset, independently of the label. We can see the distribution of this concept/relation for each class in Figure 5.4. *NeSy-IRL* approach, therefore, finds it challenging to learn rules explaining the correlation between concepts/relations and classes when such a substantial overlap exists. This focuses our attention on the concept/relation extraction step, where linking the concepts/relations and neurons may be non-representative.

We also observed that learning rules for each class separately does not improve the process, as the rules learned were overfitted to each class. When we analysed how many examples the rules covered in the test sample, it revealed a poor generalisation.

One way to deal with this problem would be to fine-tune the concepts/relations before associating them with the neurons, such as separating them into hierarchies that represent the relations between different concepts. This would help overcome

the overlapping issues and also facilitate relational reasoning about the concepts. Another point of attention was the noisy concepts/relations extracted from the model, since for the same label l_i, we noticed concepts/relations unrelated to it. For example, in the label "watching_TV", we extracted concepts/relations such as "fresh vegetables" and "clean dishes", notions that may not represent the specific context. We believe that such weak associations might be partially responsible for the poor performance.

We consider the possibility of the backpropagation on the *CNN* also interfering in the hidden unit value [85], perturbing the importance of some concepts/relations. This can lead either to the selection of the same concept/relation for various instances (overlapping) or to noisy results. It also impacts the rule-learning process as the rules generated may not generalise well. Nonetheless, our method can still extract rules qualitatively, which is an interesting result compared with the other approaches. For instance, the class "riding_a_bike" generated 65 rules which tried to represent the model's inner work. Concepts and relations, such as "a tire", "a sidewalk", "exercise clothes" and "outdoor activity" are among those learned by the rule learner that align with the label cited. Even though the rules did not generalise, the output highlights the method's potential.

These experiments revealed that we can extract declarative logic rules to bring transparency over the knowledge acquired by the model, answering Research Question *RQ5* (Section 1.5). However, evaluating the rules extracted demonstrated that the approach as of now is not yet able to obtain acceptable accuracy. Some possible aspects to improve the current approach include:

- The selection of concepts and relations that will be used to describe the dataset should be semantically related to the data, providing a solid foundation for the process. This can be achieved with a well-defined dataset, in which all the concepts and relations are associated with the images; it may require humans to create or assess the dataset.

- The approach will link the concepts and relations to the *CNN's* neurons should

guarantee sound confidence in the association between a neuron and a concept/relation.

- Improvements in the rule learning process, regarding example building and the model constraints, providing a refined method for handling noise.

Another consideration is that the penalty for a specific example is an aggregated weight over the complete context-dependent partial interpretation. Exploring strategies for weighing individual atoms in the context of a partial interpretation would be interesting.

In addition, the fact that the concepts/relations were generated by using a *LLM* can also lead to inaccurate extraction, affecting the input to the inductive rule learning process. This is due to the fact that in these experiments, our approach relies on concept/relation extraction and associates it with a hidden unit; an evolution of this process, such as an assertive association between the concepts/relations and the neurons with a strong assertiveness to the context, could improve our results. Likewise, exploring other *ILP* features in the symbolic part, as well as different background knowledge, mode bias, and penalty scoring, could lead to a better outcome.

5.5 Summary

This Chapter focuses on rule extraction, the final component of our framework aimed at providing interpretable representation of the behaviour of a trained *CNN* on classification tasks. This approach combines the learning capabilities of neural networks with the cognitive reasoning processes of symbolic rules. Inductive Logic Programming *(ILP)* is highlighted as a suitable method for generating logical rules from data, offering the advantage of requiring fewer examples compared to statistical methods. *ILP* tasks are explained using background knowledge and positive and negative examples.

To extract rules from an image classification model, this Chapter describes a method that uses Large Language Models *(LLMs)* to generate concepts and rela-

tions from dataset labels. These are then processed through a multi-modal vision-language model like *CLIP-Dissect*, which maps neurons to these concepts and relations. The activation maps from the model are used to rank the most relevant neurons for certain concepts/relations, and these rankings serve as input for the rule learning phase using *FastLAS* as *ILP* system. This method allows a deeper understanding of the model's learned representations by approximating neural activations to logical rules.

This Chapter also discusses the evaluation of the rule learner's generalisation capability. The accuracy of the rule learner is tested by comparing its performance to other baseline methods across different sample sizes. The evaluation extends to regular and top-5 accuracy, with additional analysis on the impact of sample size on model performance. The experiments indicated overall that the rule learner's effectiveness in providing understandable insights into the model's behaviour still needs improvement but we have provided some insights on how this neuro-symbolic approach to explaining the model's inner work could be improved.

Chapter 6

Conclusions

6.1 Introduction

This chapter concludes the thesis by revisiting each of the three Hypotheses we started from. It summarises the key outcomes of the research and their impact in the area of *XAI* specifically in computer vision. Each section summarises the main properties of each component of the framework: the first component aims to extract semantic concepts that represent what the pre-trained image classification model has learned during the training process (Section 6.2); the second component extracts the relations learned by the pre-trained model based on the concepts from the previous component and a knowledge graph (Section 6.3); the third and last component is tasked with extracting rules based on the concepts and relations the model has learned, aiming to provide a declarative representation of the model's decision processes (Section 6.4). We conclude this chapter by discussing the potential of this research to open up new lines of investigation in this research area (Section 6.5). We present multiple opportunities for building upon and learning from our results; we report on what we have learned from our experiments and discuss how this work could influence the direction of the field of *XAI*.

6.2 Extracting Concepts

The first step in this research (Chapter 3) involved developing a methodology to extract semantic concepts from a trained Convolutional Neural Network (*CNN*), specifically aiming to improve the model's interpretability through the identification of semantic concepts associated with individual hidden units across layers (disentangled representation). The experimental evaluation considered *CNN* models *ResNet-152, DenseNet-161, and ResNet-50*, and demonstrated how we can identify high-level concepts from the *CNN's* final layer which connect with specific classes and individual images, thus creating a bridge between broader class-level (global) and instance-specific (local) concept-driven interpretation of the model.

To adapt the approach for disentangling and extracting concepts to new datasets, we resorted to transfer learning and *Network Dissection*. This involved freezing the trained *CNN* model and replacing only the fully connected layer with a linear *SVM*, allowing us to retain existing knowledge while extending concept identification to new classification tasks. By ranking the highest activated filters or neurons per image, we prioritised concepts that reflect the most representative semantic features for each image. This technique generated a list of top-ranked concepts for each image, thus allowing us to establish a correspondence between global class-level concepts and local instance-specific attributes. The datasets *Action40* and *CIFAR-10* were used in the experimental validation of the approach to test how effectively *CNN*-extracted concepts can reflect class distinctions in the two different classification tasks.

Our method of evaluating concept importance relied on comparing local and global concept rankings for each class, yielding a precision metric that measures alignment. This metric revealed how closely *CNN*-extracted global concepts for a given class correspond with the locally ranked concepts across images within the class. Although the metric is relatively straightforward, its significance in identifying relevant global concepts was significant. This evaluation results are self-explainable, enabling a layperson to examine and validate the model's learned representations,

providing a tool for the design of concept-driven explanations.

The experimental results showed high precision when comparing global and local concept rankings, with an accuracy higher than 95% observed between the top local concepts and the highest-ranking global concepts. By examining various configurations in the number of top local and global concepts observed, the findings confirmed that *CNN* architectures like *ResNet-152* could identify consistent and meaningful concepts across classes. The model performed well with *Action40* and *CIFAR-10* datasets, where overlaps in identified concepts confirmed the semantic robustness of the extracted features.

In the design of this component, it is worth noting that the model's interpretability is currently constrained by the fixed set of concepts used by *Network Dissection* from the *Broden* dataset, which limits flexibility in concept variety and task adaptation. This consideration, along with the need to provide a richer semantic understanding of the *CNN's* decision making process, motivated the development of the second component of our framework (Chapter 4), which introduces the ability to extract relational structures based on the identified concepts.

In conclusion, this component introduces a foundation for leveraging disentangled representations as a method of interpreting *CNNs*. The component demonstrated that extracted high-level concepts support understanding of the importance of local and global semantic features and their relevance in the classification task.

6.3 Extracting Relations

In Chapter 4, we explore the use of knowledge graphs for enhancing the interpretability of *CNNs* in image classification. Specifically, we leverage the Common-sense Knowledge Graph (*CSKG*) to model a set of concepts as nodes, and edges as relations between pairs of concepts. Initially, relations between a set of input concepts are extracted from the knowledge graph by selecting only direct relations between any pair of concepts in the input set. This is done in order to mitigate noise and limit computational complexity. The selection approach ensures that the nearest edges

between pairs of concept nodes represent direct and relevant relations, making the model's interpretation more focused on meaningful semantic connections.

To assess the extent to which the deep representation (trained *CNN*) has effectively learned these semantic relations, the *Visual Genome* dataset is used as a validation tool due to its visually grounded representation of labelled relations. This dataset allows for the verification of learned relations by examining tagged relations between objects within bounding boxes. The approach focuses on images with multiple relevant concepts, using a pre-trained network to assess whether these concepts rank among the top activations in the image classification task. We also applied Named-Entity Resolution (*NER*) techniques to address mismatches in relation labels between the *CSKG* knowledge graph and the *Visual Genome* dataset, thus leveraging *NLP* tools to increase the number of relation identifications beyond mere string matching. The combination of *CSKG* and *Visual Genome* facilitates the filtering of non-relevant relations and emphasises meaningful connections, ensuring consistency with real-world visual relations.

A key focus of the component is differentiating relations between local and global concept for *Action40* and *CIFAR-10* datasets. By identifying relations that are both common and meaningful, the approach refines the *CNN's* interpretation of image classes through the *CSKG*. This filtering mechanism reveals essential local and global connections. For example, the approach can identify that a "car" is related to a "vehicle" and provide insights into the model's decision-making process and how this subsumption relation affects it. Validation of these connections using a relation coverage metric helps measure the relevance of local relations for classification tasks. For example, with high-coverage classes like "riding a bike" in *Action40* we can identify strong alignment between image-specific and class-level relations. Conversely, low-coverage classes such as "brushing_teeth" in *Action40* highlight areas where the model's conceptual understanding may be weaker, offering potential clues, such as the possible bias in the data regarding represented objects, for inspecting the model and improving the deep representation.

Results of our experiments show that the global relations learned by the model align well with common-sense knowledge in *Action40*, achieving higher relation coverage in classes with more than 30 local unique concepts, such as "fixing_a_bike" and "shooting_an_arrow". Conversely, in low-resolution datasets like *CIFAR-10*, fewer global relations emerge, suggesting limitations in concept extraction that may hinder classification performance. The experiments highlight the framework's adaptability to datasets with different pixel resolutions and concept granularities.

Ultimately, the experiment identifies several potential enhancements, such as exploring alternative concept extraction techniques, including Concept Activation Vectors (*CAV*) and *CLIP-Dissect*, and validating the *CSKG* relations against larger datasets with pixel-level annotations.

6.4 Extracting Rules

In Chapter 5, we explore the extraction of declarative rules as a way to better understand what a trained image classification model has learned, aiming to improve transparency and explainability. The rule extraction process leverages large language models (*LLMs*) to generate concept and relation sets from image labels, creating an independent and non-human-biased understanding of concepts associated with each label. Using prompts with *GPT-4*, concepts and relations are identified and used in conjunction with the *CLIP-Dissect* approach to map learned concepts to specific neurons. This technique bypasses the traditional reliance on human annotation and paves the way for the automated analysis of model knowledge by associating features directly with neuron activations.

To effectively link concepts with neurons, a similarity function computes the association strength between extracted concepts and neuron activations in a trained *CNN*. Each *CNN* layer contains neurons with activations that represent how well a particular concept is captured. A selected *CNN* model processes these activations, identifying the most relevant neurons based on similarity scores, which are subsequently ranked to determine key concepts. This ranking process results in a dataset

that reflects the highest-ranked concepts for each image, which becomes the input for rule learning using inductive logic programming (*ILP*).

The *FastLAS* system is used for inductive rule learning, by integrating background knowledge and language biases that constrain the search space for generating logical rules. Predicates, such as "label" and "concept-relation", are represented in this background knowledge, specifying that only one class label can be selected for each image. Using *FastLAS*, symbolic rules are generated that link concepts and relations to specific class labels, allowing the logical patterns learned by the model to be systematically evaluated. This rule-generation process aims to identify the structure of knowledge learned by the model and provide an alternative to conventional final-layer classification methods that can be used directly for generating explanation traces.

Experiments were conducted on two datasets, namely *CIFAR-10* and *Action40*, to evaluate the rule-extraction component. These experiments measured the accuracy of the generated rules and compared performance with other baselines. The experiments highlight challenges with more complex datasets such as *Action40*, where overlapping concepts may directly impact rule accuracy. An extended top-5 accuracy metric was adopted to measure accuracy in cases where multiple predictions were made, helping to capture scenarios in which the top result may not be correct. Still, the accuracy was lower than 10%, demonstrating the challenges of this approach.

Experimental results reveal the strengths and limitations of rule-based learning in *AI* interpretability. For both datasets (*CIFAR-10* and *Action40*), the approach demonstrates a significant concept overlap among classes that challenges the model's performance. Despite these challenges, rule-based learning can present a novel approach to understanding *AI* model learning processes, especially in scenarios where training data is limited or biased, offering an alternative pathway to purely statistical approaches.

6.5 Challenges and Opportunities ahead

In this thesis, we propose a novel framework to extract semantic concepts, relations, and rules that can approximate the inner workings of a pre-trained *CNN* classification model for a computer vision task. We claim that each component in our framework can offer a deeper understanding of the decision-making process of *CNNs*, thus increasing transparency for domain experts. This is particularly important when neural models used in computer vision make mistakes, but it can also be a key enabler for the effective combination of learning (from data) and reasoning (about a domain).

In the remainder of this section we discuss the open challenges and research opportunities when it comes to exploring symbolic methods for neural networks. This is a field of research which is recently regaining attention and is known as *Neurosymbolic AI* [40], and we are particularly interested in opportunities for *Neurosymbolic* integration that can enhance explainability and trust in black-box models [87].

Human-centred explainability

The ability to enhance the validation procedure when it comes to explainability requires the involvement of human experts in the process. One interesting avenue for further investigation is on the introduction of human-in-the-loop approaches to explaining black-box models, whereby the model is able to incorporate expert feedback to assess explanations and adjust the explanation generation process accordingly. Such techniques have been extensively used in the realm of machine learning to augment model training, validation, and testing, mostly focusing on developing adaptive strategies for annotation and labelling. The application of these methods in the context of *XAI*, particularly in terms of explanation validation and the adaptive generation of explanations, is still underinvestigated.

In other words, we do not only want the accuracy to be high, but we also want to have means to validate that the model produces the correct answer for the right reason. Human feedback can be harnessed and integrated into our approach to

verifying concepts/relations and rules approximating the decision-making process of the trained *CNN*. For instance, adjusting the ranking of concepts or correcting extracted rules and relations based on human knowledge can reveal significant semantic aspects that have not been adequately captured in the deep representation. This corrective knowledge should then be injected back into the training process to learn a better representation and, as a result, produce better explanations.

Despite acknowledging the importance of this step, we have not conducted a human evaluation in this work due to the early stage of research advancements in this area. In other words, the need to ensure (and validate) soundness and robustness in the extraction of high-level concepts learned by the model compelled us to focus on concept extraction first, to pave the way for designing robust human evaluation protocols. Furthermore, there is still an issue with the definition and adoption of universal evaluation metrics, making this task even more challenging at this stage of research. These metrics should consider various aspects of the evaluators (including their subjectivity) to enhance the validation process from a human perspective, and this area has only recently become a focal point of active research.

Generic vs. Specific Concepts and Relations Extraction

One of the key aspects of the proposed framework is the semantic concepts and relations available to support the approach. Currently, the concepts on which the concept extraction component relies are derived from the *Broden* dataset; similarly, the relations are based on the *CSKG* knowledge graph. Although these knowledge sources are a suitable starting point for our investigation, they introduce some limitations as they constrain the type concepts and relations produced by the framework. The specific type of concepts present in the *Broden* dataset can prevent our concept extraction component from learning concepts that have not been part of the disentanglement process.

Similarly, the use of *CSKG* as Knowledge Graph does not cover all possible relations between concepts (specifically when we are in a well defined target domain

such as health). Similar considerations can be made when it comes the use of the *Visual Genome* dataset to validate the relations learned by the model, as some relations might be missing. In order to broaden the applicability of our approach, it is necessary to see how it would perform when disentangled concepts as well as knowledge graphs used in the respective components of our framework, are domain specific, considering that this study focused on a general domain. There is still a lack of baseline approaches in specific fields and therefore researchers tend to consider general domains, but we believe there is a missed opportunity here to create methods for concept disentanglement and relation extraction that work well in specific domains.

Concept extraction techniques

One limitation of our method is its reliance on *Network Dissection* for concept extraction on which all other components depend. It would be interesting to explore the performance of recent alternative methods for disentangling concepts, such as Concept Activation Vectors [61], or multi-modal approaches, such as *CLIP-Dissect* [93]. Leveraging the strengths of each approach and integrating them could enhance the technique.

In Chapter 5, we used the *CLIP-Dissect* approach to explore how this could potentially affect the results obtained in our framework. Our experiments revealed the possibility of extracting concepts without relying on a predefined set of concepts. However, a deeper investigation is needed to verify how reliably *LLMs* can generate such information, and how the use of external knowledge could improve the semantic association between the concepts/relation and the neurons.

Rule Learning

The experiments presented in Chapter 5 show that our method has a wide margin for improvements on the rule learning process. Specifically, the design of better methods that can handle noise is required. Currently, the noise is an aggregated

weight over the complete context-dependent partial interpretation. Exploring methods for weighing individual atoms in the context of a partial interpretation would be interesting and possibly improve the results. The fact that the concepts were generated by a *LLM* can also lead to noise in the concept extraction process, thus affecting the input to the inductive rule learning component of the framework. A more extensive analysis and comparison of other *LLM* approaches, such as "Gemini" [118] and "Llama" [121], could yield more representative concepts.

In terms of how well the extracted rules approximate the *CNN's* decision making process, we believe that the use of more refined background knowledge and the integration of other *ILP* features in the symbolic part, such as i) different predicates and bias models, ii) using more complex rules with probabilities, or iii) implementing a different learning approach, such as learning from entailment or satisfiability [27], could also bring improved outcomes. Due to time constraints, we were unable to experiment with these different approaches. Additionally, enhancing background knowledge with external data could refine the rule-learning process.

Evaluation Metrics and Benchmarks

The lack of robust qualitative metrics for *XAI* approaches that resort to the combination of semantic and neural elements remains an ongoing challenge in the research community. This is partially due to the fact that, despite several *XAI* approaches having been developed, there is often a lack of datasets and benchmarks that include the ground truth of explanations to fully validate the models [4, 89, 102]. Even when producing relevant outputs from what the model has learned, most evaluation metrics are based on quantitative analysis. Measuring how transparent or interpretable an explanation is remains subjective (as explanations are for humans and should be assessed based on their utility to humans) and domain-specific, and there are significant opportunities for the research community in this area [106]. One way to overcome this is to use humans to evaluate explanations, employing metrics that measure how reasonable such an explanation is, how the explanation relates to

common-sense knowledge, and how confident the evaluator is regarding the explanation generated. This work initially focused on generating the high-level features needed to comprehend the trained model, thereby aiding future human evaluation efforts.

Even though we do not generate explanations as such, the ability to validate the outcome of each component (concepts, relations and rules) calls for additional qualitative human-centred metrics that could analyse how well a set of concepts/relations represents a given instance or class. Regarding concepts (the outcome of the first component), we measure the intersection between the local and global concepts for a specific class. However, no robust and absolute metric exists to verify how well such concepts represent the class, as this should go beyond the available data. Similarly, in terms of relations (the output of the second component), we focus on measuring the consistency between common-sense knowledge graph relations and the concepts derived from the first component and the visual dataset used for validation. There is an opportunity here to explore the use of additional metrics that can help identify the suitability of different knowledge graphs for a specific task before looking into the consistency between a given knowledge graph and the extracted concepts. In this study, we focused on using a generic domain to establish a ground truth approach.

In the rule learning process, the validation of the quality of the extracted rules is purely quantitative, and it relies on the rule length or the number of rules generated, alongside the rule coverage concerning the model output. These metrics do not capture the qualitative nature of the evaluation, as they cannot assess how well the rules align with human understanding and decision-making for the specific task. The design of more robust quantitative metrics, benchmarks, and ground truth to assess the generated explanations presents great opportunities for significant advances in the field of *XAI*.

Appendices

Appendix A

Publications on Work from this Thesis

A.1 Conference Publications

- Ferreira dos Santos, E., Mileo, A. (2022). From Disentangled Representation to Concept Ranking: Interpreting Deep Representations in Image Classification Tasks. In: Koprinska, I., et al. Machine Learning and Principles and Practice of Knowledge Discovery in Databases. ECML PKDD 2022. Communications in Computer and Information Science, vol 1752. Springer, Cham. https://doi.org/10.1007/978-3-031-23618-1_22

- Ferreira dos Santos, E., Mileo, A. (2024). Towards Understanding Deep Representations in CNN: from Concepts to Relations Extraction via Knowledge Graphs. In: AICS'24: 32nd Irish Conference on Artificial Intelligence and Cognitive Science, December 09–10, 2024

A.2 PhD Consortium

- Ferreira dos Santos, E. (2023). Concepts, Relations and Rules Extraction from Trained Convolutional Neural Networks: A Framework for Explainability. In:

26th European Conference on Artificial Intelligence, September 30 – October 4, 2023

Appendix B

Experimental details

B.1 Datasets

B.1.1 Broden dataset

The *Broden* dataset [141] is a diversified picture dataset used in computer vision research to analyse visual scenes. It includes thousands of pictures from diverse situations and extensive annotations for items, parts, materials, and scenarios. Following [141], this dataset can be used to associate filters and high-level concepts.

B.1.2 Action40

The *Action40* dataset [131] is a collection of images used in computer vision research for action recognition. It comprised 40 action classes and was used to study what the pre-trained model knows when applied to diverse data sets via transfer learning. Tables B.1 and B.2 present the data distribution between train and test for each class. Tables B.3, B.5 and B.7 present the top 10 local concepts per class for each architecture, while Tables B.4, B.6 and B.8 present the top 10 global concepts per class for each architecture.

Unique concepts

- ResNet-152: 'arm', 'hand', 'chair', 'hair', 'plant', 'airplane', 'person', 'shop window', 'bed', 'bus', 'plate', 'car', 'table', 'washer', 'tvmonitor', 'body', 'dog', 'motorbike', 'road', 'truck', 'seat', 'coach', 'bird', 'plaything', 'grass', 'fence', 'leg', 'court', 'food', 'sea', 'rock', 'apparel', 'headboard', 'head', 'shelf', 'toilet', 'muzzle', 'skyscraper', 'metal', 'pool table', 'slot machine', 'seat cushion', 'cow', 'bedclothes', 'stage', 'bottle', 'boat', 'curtain', 'tennis court', 'fabric', 'horse', 'dome', 'skin', 'wheel', 'sky', 'footboard', 'pot', 'train', 'building', 'torso', 'tree', 'headlight', 'cat', 'ear', 'black-c', 'windowpane', 'orange-c', 'ceiling', 'red-c', 'frame', 'paper', 'screen', 'labyrinth', 'concrete', 'house', 'water', 'pole', 'pane', 'stove', 'flower', 'pottedplant', 'palm', 'base', 'sand', 'field', 'snow', 'signboard', 'sheep', 'purple-c', 'bathtub', 'bridge', 'sink', 'stern', 'book', 'ball', 'bookcase', 'sidewalk', 'drawer', 'earth', 'bicycle', 'yellow-c', 'crosswalk', 'wood', 'mountain', 'plastic-clear', 'chandelier', 'fireplace', 'cradle', 'painting', 'desk', 'cage', 'floor', 'carpet', 'tent', 'silver screen', 'shade', 'monitor', 'roof', 'refrigerator', 'door', 'railing', 'wardrobe', 'machine', 'cushion', 'oven', 'case', 'leather', 'brick', 'waterfall', 'ceramic', 'lid', 'manhole', 'track', 'mirror', 'tile', 'wing', 'bench', 'keyboard', 'top', 'barrel', 'cabinet', 'river', 'carousel', 'box', 'column', 'stairway', 'back pillow', 'arcades', 'arcade machine', 'pillow', 'laptop', 'towel'

- ResNet-50: 'table', 'person', 'chair', 'field', 'seat', 'motorbike', 'pool table', 'work surface', 'arm', 'toilet', 'bottle', 'stove', 'bed', 'pottedplant', 'airplane', 'car', 'road', 'tree', 'building', 'truck', 'body', 'boat', 'court', 'dog', 'cat', 'bird', 'food', 'leg', 'red-c', 'horse', 'muzzle', 'hand', 'shelf', 'arcade machine', 'washer', 'box', 'grass', 'shop window', 'coach', 'head', 'house', 'hair', 'train', 'cow', 'torso', 'leather', 'mountain', 'wallpaper', 'track', 'fabric', 'tile', 'wood', 'fence', 'water', 'swimming pool', 'skin', 'sea', 'sky', 'screen', 'tent', 'cradle', 'sand', 'shade', 'wheel', 'ear', 'carpet', 'top', 'case', 'sink', 'keyboard', 'aquarium', 'purple-c', 'rock', 'bridge', 'door', 'headboard', 'windowpane', 'plant', 'bed-

clothes', 'slot machine', 'brick', 'snow', 'sidewalk', 'waterfall', 'bathtub', 'book', 'curtain', 'barrel', 'bicycle', 'pink-c', 'ceiling', 'bookcase', 'carousel', 'flower', 'bus', 'refrigerator', 'plaything', 'footboard', 'black-c', 'labyrinth', 'plate', 'sheep', 'skyscraper', 'apparel', 'fur', 'seat cushion', 'yellow-c', 'pot', 'metal', 'pane', 'stairway', 'railing', 'chandelier', 'lid', 'roof', 'desk', 'paper', 'neck', 'ball', 'earth', 'headlight', 'floor', 'dome', 'aqueduct', 'river', 'pole', 'aircraft carrier', 'painting', 'smoke', 'drawer', 'concrete', 'back pillow', 'cabinet', 'mirror', 'fireplace', 'monitor', 'fire escape', 'white-c', 'wing', 'stern', 'plastic-opaque', 'ceramic', 'crosswalk', 'shaft', 'sofa', 'signboard', 'computer', 'nose', 'television', 'tvmonitor', 'animal'

- DenseNet-161: 'person', 'airplane', 'seat', 'sofa', 'painting', 'sea', 'signboard', 'table', 'house', 'dog', 'plate', 'hair', 'tree', 'train', 'food', 'head', 'cow', 'bicycle', 'screen', 'body', 'bed', 'court', 'bus', 'carpet', 'road', 'swimming pool', 'building', 'toilet', 'muzzle', 'bookcase', 'wardrobe', 'motorbike', 'car', 'windowpane', 'curtain', 'bridge', 'cat', 'carousel', 'sidewalk', 'horse', 'footboard', 'drawer', 'bird', 'torso', 'tile', 'sink', 'metal', 'grass', 'skyscraper', 'stove', 'field', 'ceiling', 'leg', 'coach', 'lamp', 'stairway', 'leather', 'mountain', 'roof', 'bottle', 'sheep', 'hand', 'sand', 'red-c', 'fountain', 'barrel', 'shelf', 'pool table', 'rock', 'brick', 'wing', 'case', 'ball', 'fence', 'pane', 'plant', 'snow', 'tent', 'refrigerator', 'arm', 'shop window', 'balcony', 'chair', 'cradle', 'ear', 'purple-c', 'pot', 'ceramic', 'labyrinth', 'canopy', 'orange-c', 'stern', 'bathtub', 'chandelier', 'flower', 'pink-c', 'wheel', 'arcade machine', 'fabric', 'black-c', 'seat cushion', 'apparel', 'water', 'railing', 'cabinet', 'washer', 'aircraft carrier', 'lid', 'truck', 'book', 'shade', 'fireplace', 'laptop', 'concrete', 'wood', 'yellow-c', 'sky', 'white-c', 'boat', 'desk', 'waterfall', 'frame', 'box', 'track', 'clouds', 'slot machine', 'blue-c', 'keyboard'

	Train	Test
applauding	100	184
blowing_bubbles	100	159
brushing_teeth	100	100
cleaning_the_floor	100	112
climbing	100	195
cooking	100	188
cutting_trees	100	103
cutting_vegetables	100	89
drinking	100	156
feeding_a_horse	100	187
fishing	100	173
fixing_a_bike	100	128
fixing_a_car	100	151
gardening	100	99
holding_an_umbrella	100	192
jumping	100	195
looking_through_a_microscope	100	91
looking_through_a_telescope	100	103
phoning	100	159

Table B.1: Action40 Dataset Distribution

	Train	Test
playing_guitar	100	189
playing_violin	100	160
pouring_liquid	100	100
pushing_a_cart	100	135
reading	100	145
riding_a_bike	100	193
riding_a_horse	100	196
rowing_a_boat	100	85
running	100	151
shooting_an_arrow	100	114
smoking	100	141
taking_photos	100	97
texting_message	100	93
throwing_frisby	100	102
using_a_computer	100	130
walking_the_dog	100	193
washing_dishes	100	82
watching_TV	100	123
waving_hands	100	110
writing_on_a_board	100	83
writing_on_a_book	100	146

Table B.2: Action40 Dataset Distribution

Class	Concepts
applauding	[person, hand, bird, hair, arm, plant, food, court, grass, road]
blowing_bubbles	[hand, person, hair, pool table, bottle, food, road, plaything, shop window, dog]
brushing_teeth	[person, hand, dog, bed, hair, headboard, stove, body, arm, ball]
cleaning_the_floor	[leg, pole, stove, car, bus, person, hand, headboard, bed, court]
climbing	[hand, person, leg, rock, bird, pool table, mountain, building, plant, tree]
cooking	[stove, food, hand, arm, person, plant, house, table, pot, case]
cutting_trees	[house, tree, plant, bird, person, bicycle, hand, motorbike, snow, food]
cutting_vegetables	[hand, food, stove, arm, person, plate, plant, table, case, metal]
drinking	[hand, person, arm, dog, hair, bed, food, bottle, leg, body]
feeding_a_horse	[horse, house, grass, plant, dog, tree, cow, hand, earth, fence]
fishing	[water, dog, airplane, sea, person, bus, washer, plant, tree, hand]
fixing_a_bike	[bicycle, wheel, plant, road, house, hand, motorbike, person, mountain, arm]
fixing_a_car	[plant, road, car, house, hand, motorbike, person, case, airplane, bus]
gardening	[plant, house, grass, person, bed, hand, tree, arm, stove, food]
holding_an_umbrella	[road, tent, person, plant, ball, muzzle, hand, roof, train, dog]
jumping	[leg, person, bird, hand, road, plant, sky, sea, water, tree]
looking_through_a_microscope	[hand, person, metal, arm, tennis court, shelf, plant, table, monitor, stove]
looking_through_a_telescope	[hand, plant, person, house, airplane, skyscraper, bicycle, monitor, tree, dog]
phoning	[person, hand, hair, road, bed, plant, apparel, bus, arm, leg]
playing_guitar	[skyscraper, bus, wheel, stage, hair, house, person, torso, hand, plant]
playing_violin	[person, skyscraper, case, dog, food, hand, pole, plant, bus, house]
pouring_liquid	[hand, person, arm, stove, food, plant, bottle, table, metal, hair]
pushing_a_cart	[road, chair, cradle, house, plant, airplane, food, book, sidewalk, person]
reading	[hand, bed, person, hair, book, plant, arm, road, bookcase, chair]
riding_a_bike	[bicycle, wheel, motorbike, road, mountain, plant, person, hand, tree, house]
riding_a_horse	[horse, grass, tree, plant, field, earth, sea, dog, house, cow]
rowing_a_boat	[water, airplane, sea, boat, person, body, pool table, plant, house, seat cushion]
running	[road, person, leg, grass, sea, tree, bird, boat, arm, hand]
shooting_an_arrow	[hand, bus, washer, person, plant, tree, house, wheel, horse, bicycle]
smoking	[hand, person, hair, arm, bed, road, plant, leg, food, dog]
taking_photos	[hand, person, bird, shelf, road, dog, plant, bicycle, food, horse]
texting_message	[hand, person, arm, hair, plant, bed, metal, road, stove, food]
throwing_frisby	[grass, horse, tree, bird, pool table, leg, food, hand, person, sea]
using_a_computer	[screen, hand, book, monitor, bed, table, headboard, arm, desk, plant]

Class	Concepts
walking_the_dog	[dog, bed, grass, road, person, horse, house, leg, sidewalk, hand]
washing_dishes	[stove, hand, arm, person, plant, food, headboard, metal, table, bed]
watching_TV	[screen, bed, tvmonitor, hand, seat cushion, plant, stove, desk, person, headboard]
waving_hands	[person, hand, plant, bed, leg, road, bird, hair, arm, house]
writing_on_a_board	[hand, person, stage, tennis court, headboard, hair, paper, court, bed, train]
writing_on_a_book	[hand, hair, arm, book, person, table, paper, stove, screen, bed]

Table B.3: Top 10 Local Concepts - Action40/ResNet-152

B.1.3 Visual Genome

The *Visual Genome* [65] dataset is a large-scale picture dataset created for visual scene interpretation in computer vision research. It has object bounding boxes, object names, object properties, object connections, scene attributes, and other fine-grained annotations, enabling extensive analysis of visual image information. It includes around 42,000 distinct relations marked [1], with an estimated 21 associations per picture. This study employed it to determine which relations were learned from the model, considering the concepts already learned.

B.1.4 CSKG

The *CSKG* (Commonsense Knowledge Graph) dataset [57] is a large-scale knowledge graph that captures common sense information about the world. It is employed in natural language processing research for language interpretation and reasoning problems because it provides organised information about concepts, entities, and connections. *CSKG* has around 2,087,000 concepts with 58 distinct relations. This dataset investigated the probable common-sense relation between the two concepts learned.

[1] https://ango.ai/visual-genome-introduction/

Class	Concept
applauding	[hand, hair, bird, grass, person, stage, airplane, court]
blowing_bubbles	[hand, bottle, shop window, pool table, person, plaything, headlight]
brushing_teeth	[dog, body, hand, pool table, stern, horse, muzzle, bottle]
cleaning_the_floor	[leg, bus, pole, body, sea, car, headboard, wood, court, crosswalk]
climbing	[pool table, hand, building, sea, ceiling, rock, tree, mountain, train]
cooking	[stove, food, barrel, table, dog, torso, house, motorbike]
cutting_trees	[snow, tree, bird, house, plant, motorbike, hand, bicycle]
cutting_vegetables	[pottedplant, stove, plate, pool table, food, arm]
drinking	[hand, muzzle, body, bottle, food, dog]
feeding_a_horse	[hair, fence, boat, sheep, bus, cow, grass, horse, dog, head]
fishing	[bus, washer, water, dog, person, sea, tree]
fixing_a_bike	[wheel, bicycle, person, floor, metal, sidewalk]
fixing_a_car	[car, motorbike, body, toilet, train]
gardening	[plant, earth, grass, pottedplant, train, flower]
holding_an_umbrella	[tent, dog, roof, muzzle, carpet]
jumping	[bird, footboard, person, hand, leg, pool table, airplane]
looking_through_a_microscope	[monitor, hand, shelf, tennis court, metal, airplane, sink, arm, bottle, bicycle]
looking_through_a_telescope	[hand, skyscraper, body, dog, monitor, car, sky]
phoning	[screen, bicycle, pool table, food, body, bottle, apparel, pole, dog]
playing_guitar	[skyscraper, bus, shelf, cage, muzzle, wheel, stage, torso, house]
playing_violin	[case, person, dog, food, cat, motorbike, horse]
pouring_liquid	[person, bottle, waterfall, plastic-clear, windowpane, headlight, hand, food, body]
pushing_a_cart	[airplane, cradle, road, chair, food, wheel, cat, box, drawer]
reading	[book, table, paper, bus, tent, chair, body, screen, palm, bed]
riding_a_bike	[bicycle, wheel, motorbike, bed, bathtub]
riding_a_horse	[horse, grass, train, dog, toilet, sea, cow]
rowing_a_boat	[water, body, house, boat, person, airplane]
running	[road, person, body, leg, apparel]
shooting_an_arrow	[washer, bus, wheel, stern, hand, tree, person, horse, bird]
smoking	[hand, plate, airplane, screen, hair, person, concrete, horse, car]
taking_photos	[bicycle, hand, skyscraper, dog, person, washer, screen]
texting_message	[hand, person, hair, bus, orange-c, slot machine, ceiling, sky]
throwing_frisby	[grass, arm, washer, sea, leg, tree, pool table, hand, horse]
using_a_computer	[screen, monitor, seat, keyboard, track, roof, tvmonitor, hair]

Class	Concept
walking_the_dog	[person, dog, sidewalk, grass, road, bathtub, bed]
washing_dishes	[airplane, stove, cradle, sink, bathtub]
watching_TV	[screen, tvmonitor, seat cushion, silver screen, cabinet]
waving_hands	[person, hand, car, tree, sky]
writing_on_a_board	[paper, stage, roof, pool table, food, cow, tennis court, refrigerator, train]
writing_on_a_book	[airplane, paper, boat, screen, hand, apparel, hair]

Table B.4: Top 10 Global Concepts - Action40/ResNet-152

B.1.5 CIFAR-10

The *CIFAR-10* [67] dataset is a widely used benchmark dataset in machine learning, particularly for image classification tasks. It consists of 60,000 32x32 colour images across 10 different classes: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. The dataset is split into 50,000 training images and 10,000 test images. Each class contains 6,000 images, and the dataset is balanced. *CIFAR-10* is often used to evaluate the performance of various image recognition algorithms, including convolutional neural networks (*CNNs*). Table B.9 presents the data distribution between train and test for each class. Tables B.10, B.12 and B.14 present the top 10 local concepts per architecture, while Tables B.11, B.13 and B.15 present the top 10 global concepts per architecture.

Unique Concepts

- ResNet-152: 'seat cushion', 'stove', 'paper', 'smoke', 'water', 'bird', 'body', 'person', 'plant', 'shelf', 'boat', 'airplane', 'pool table', 'sea', 'cat', 'sky', 'torso', 'hair', 'wing', 'pottedplant', 'headboard', 'tree', 'grass', 'tent', 'tvmonitor', 'dog', 'shade', 'car', 'train', 'seat', 'sink', 'bathtub', 'hand', 'field', 'wheel', 'motorbike', 'white-c', 'frame', 'cradle', 'stern', 'fireplace', 'orange-c', 'skyscraper', 'table', 'food', 'footboard', 'stage', 'building', 'snow', 'screen', 'chandelier', 'fence', 'earth', 'road', 'apparel', 'silver screen', 'coach', 'rock', 'concrete',

Class	Concepts
applauding	[person, field, pool table, dog, seat, bird, grass, leg, torso, hair]
blowing_bubbles	[person, field, plaything, hair, building, grass, tree, bird, plant, pool table]
brushing_teeth	[person, field, plaything, hand, stove, muzzle, hair, headboard, toilet, head]
cleaning_the_floor	[person, field, leg, car, bird, hand, pool table, headboard, stove, toilet]
climbing	[person, field, rock, mountain, boat, brick, leg, building, pool table, toilet]
cooking	[field, stove, case, pool table, person, plate, pot, work surface, food, arm]
cutting_trees	[field, tree, rock, plant, person, mountain, earth, motorbike, car, grass]
cutting_vegetables	[stove, field, case, person, pool table, arm, food, plate, work surface, pot]
drinking	[field, person, pool table, hair, hand, muzzle, arm, bottle, body, stove]
feeding_a_horse	[field, horse, grass, tree, person, dog, plant, fence, sky, boat]
fishing	[water, sea, field, boat, person, sky, tree, grass, metal, mountain]
fixing_a_bike	[bicycle, field, person, wheel, building, motorbike, arm, mountain, grass, pool table]
fixing_a_car	[field, car, tree, road, person, motorbike, building, pool table, boat, sea]
gardening	[plant, field, tree, person, grass, rock, stove, arm, earth, pot]
holding_an_umbrella	[building, person, field, pool table, flower, road, washer, tree, tent, roof]
jumping	[person, field, leg, tree, grass, sea, plant, sky, bird, building]
looking_through_a_microscope	[field, person, airplane, table, work surface, toilet, case, hair, pool table, book]
looking_through_a_telescope	[field, person, tree, pool table, sea, airplane, leg, grass, boat, body]
phoning	[person, field, hair, building, leg, pool table, headboard, muzzle, screen, arm]
playing_guitar	[skyscraper, person, field, leg, cradle, pool table, hair, muzzle, bird, dog]
playing_violin	[field, bird, leg, body, person, hand, dog, skyscraper, shop window, pool table]
pouring_liquid	[field, person, pool table, stove, arm, case, plate, table, hair, work surface]
pushing_a_cart	[field, building, person, road, tree, cradle, pool table, fence, dog, case]
reading	[field, person, hair, book, arm, building, pool table, bed, dog, top]
riding_a_bike	[bicycle, field, motorbike, building, person, mountain, road, tree, grass, wheel]
riding_a_horse	[field, grass, horse, tree, person, ceiling, sky, dog, plant, motorbike]
rowing_a_boat	[water, boat, sea, field, person, pool table, airplane, grass, motorbike, tree]
running	[field, person, tree, leg, sea, grass, road, arm, building, sand]
shooting_an_arrow	[field, person, sky, metal, grass, tree, horse, boat, plant, bridge]
smoking	[person, field, hair, building, arm, bird, pool table, leg, hand, motorbike]
taking_photos	[field, person, leg, building, hand, grass, sea, motorbike, pool table, bird]
texting_message	[person, field, hair, pool table, arm, building, stove, table, headboard, hand]
throwing_frisby	[grass, field, person, tree, leg, court, dog, pool table, arm, wood]
using_a_computer	[field, screen, top, desk, book, hair, bed, person, headboard, airplane]

Class	Concepts
walking_the_dog	[dog, field, person, tree, grass, building, leg, cat, plant, road]
washing_dishes	[stove, field, person, work surface, pool table, headboard, arm, case, pot, toilet]
watching_TV	[field, screen, headboard, pool table, top, person, stove, bed, hair, television]
waving_hands	[field, person, tree, pool table, building, boat, leg, grass, dog, plant]
writing_on_a_board	[person, field, hair, pool table, toilet, court, work surface, headboard, screen, body]
writing_on_a_book	[field, person, cow, hair, arm, pool table, table, top, screen, desk]

Table B.5: Top 10 Local Concepts - Action40/ResNet-50

'cushion', 'red-c', 'house', 'bus', 'waterfall', 'mountain', 'horse', 'ceramic', 'muzzle', 'yellow-c', 'bicycle', 'sand', 'top', 'ear', 'carpet', 'track', 'leg', 'bridge', 'head', 'toilet', 'bookcase', 'arm', 'windowpane', 'monitor', 'bottle', 'tennis court', 'flower', 'black-c', 'bed', 'metal', 'crosswalk', 'labyrinth', 'palm', 'chair', 'stairway', 'cow', 'washer', 'plaything', 'case', 'plate', 'signboard', 'pot', 'railing', 'fabric', 'drawer', 'roof', 'ceiling', 'truck', 'bench', 'sheep', 'desk', 'plastic-clear', 'grand piano', 'pole', 'bedclothes', 'court', 'leather', 'sidewalk', 'barrel', 'slot machine', 'pillow', 'headlight', 'dome', 'keyboard', 'carousel', 'box', 'shop window', 'floor', 'lid', 'book', 'purple-c', 'refrigerator', 'sofa', 'ball', 'wardrobe', 'skin', 'back pillow', 'painting', 'cage', 'cabinet', 'pane', 'tile', 'river', 'arcades', 'arcade machine', 'paw', 'curtain', 'brick', 'door', 'towel', 'wood', 'oven', 'machine', 'column'

- ResNet-50: 'dog', 'cabinet', 'pool table', 'wing', 'hair', 'ceramic', 'sea', 'wheel', 'water', 'screen', 'ear', 'tree', 'person', 'house', 'airplane', 'white-c', 'torso', 'sky', 'pot', 'hand', 'horse', 'plant', 'stern', 'lid', 'grass', 'motorbike', 'seat cushion', 'boat', 'road', 'bird', 'sheep', 'body', 'rock', 'shade', 'sink', 'railing', 'coach', 'head', 'field', 'stove', 'cushion', 'car', 'red-c', 'fence', 'chair', 'food', 'bridge', 'cat', 'mountain', 'pottedplant', 'skyscraper', 'leather', 'headboard', 'bed', 'fur', 'bus', 'seat', 'earth', 'toilet', 'paper', 'flower', 'bottle', 'leg', 'plate', 'bathtub', 'river', 'ceiling', 'muzzle', 'washer', 'building', 'refrigerator', 'carpet', 'crosswalk', 'train', 'drawer', 'arm', 'sand', 'plaything', 'stairway', 'tile',

Class	Concept
applauding	[coach, person, hand, house, bed, rock, seat, ear, horse]
blowing_bubbles	[plaything, plate, ceiling, headlight, pool table, cat, flower, motorbike, bird]
brushing_teeth	[headboard, muzzle, body, door, hand, house]
cleaning_the_floor	[car, road, bird, floor, sheep, dog, horse, leg, sidewalk]
climbing	[mountain, brick, food, toilet, motorbike, carousel, sky, tent, hand]
cooking	[stove, plate, person, case, food, metal, shade]
cutting_trees	[motorbike, rock, car, food, field, person, tree, dog]
cutting_vegetables	[stove, bottle, food, chair, arm, grass, pool table]
drinking	[boat, hand, black-c, bed, body, bottle, roof]
feeding_a_horse	[horse, airplane, person, field, muzzle, grass, dog]
fishing	[metal, sea, bridge, sky, grass, car, water, boat]
fixing_a_bike	[bicycle, wheel, pool table, coach, tile]
fixing_a_car	[car, pool table, train, wheel, airplane, motorbike, ceiling]
gardening	[plant, food, bus, flower, person]
holding_an_umbrella	[dog, flower, roof, pool table, mountain, tent, building, wing]
jumping	[carpet, leg, bird, person, dog, railing, airplane, wheel, field]
looking_through_a_microscope	[food, book, airplane, toilet, sheep, flower, hair, motorbike, shelf]
looking_through_a_telescope	[muzzle, sheep, airplane, sky, hand, leg, metal, body, dog, house]
phoning	[screen, dog, rock, muzzle, water, door, fabric, boat, bicycle]
playing_guitar	[skyscraper, sheep, cradle, body, bird, table, hair, cat, muzzle, lid]
playing_violin	[dog, bird, body, hand, food, motorbike, car, table, boat]
pouring_liquid	[motorbike, red-c, pool table, wood, person, plastic-opaque, sea, track]
pushing_a_cart	[road, wheel, fence, floor, carpet, field, shelf, book]
reading	[book, red-c, court, food, cat, truck, dog, hand, muzzle]
riding_a_bike	[bicycle, road, motorbike, earth, arm, mountain]
riding_a_horse	[horse, body, grass, person, leather, house]
rowing_a_boat	[water, boat, pool table, leg, person, motorbike, grass]
running	[wood, road, coach, tree, person, sand, motorbike, chair, dog, leg]
shooting_an_arrow	[metal, sky, bridge, car, grass, shop window, horse, person, cow, motorbike]
smoking	[motorbike, head, bicycle, labyrinth, stove, bird, plant, torso, bottle]
taking_photos	[sea, hand, leg, toilet, grass, head, motorbike, washer, train]
texting_message	[bus, pool table, hair, coach, bird, airplane, fabric]
throwing_frisby	[court, plaything, bird, person, water, shade, sea, headboard, grass]

Class	Concept
using_a_computer	[pot, carpet, computer, field, book, grass, bed, desk, fabric, screen]
walking_the_dog	[dog, person, bird, building, leg, sheep]
washing_dishes	[toilet, bird, water, work surface, arm, pool table, boat, motorbike, bathtub, purple-c]
watching_TV	[screen, bed, television, drawer, headboard, building]
waving_hands	[coach, bird, hand, person, cat, court, tree, muzzle, horse]
writing_on_a_board	[cat, water, body, bird, train, wing, person, waterfall]
writing_on_a_book	[screen, cow, drawer, pool table, skin, hand, boat, bed, table]

Table B.6: Top 10 Global Concepts - Action40/ResNet-50

'keyboard', 'carousel', 'bicycle', 'tent', 'court', 'smoke', 'truck', 'door', 'fireplace', 'pole', 'windowpane', 'yellow-c', 'table', 'metal', 'cradle', 'desk', 'television', 'floor', 'wood', 'apparel', 'arcade machine', 'shelf', 'cow', 'shop window', 'snow', 'bedclothes', 'footboard', 'roof', 'case', 'fire escape', 'book', 'painting', 'slot machine', 'nose', 'mirror', 'waterfall', 'bookcase', 'chandelier', 'sidewalk', 'top', 'purple-c', 'black-c', 'pane', 'pink-c', 'wallpaper', 'barrel', 'lamp', 'back pillow', 'curtain', 'plastic-opaque', 'dome', 'headlight', 'aquarium', 'labyrinth', 'skin', 'track', 'aqueduct', 'ball', 'neck', 'brick', 'fabric', 'signboard', 'work surface', 'buffet', 'aircraft carrier', 'swimming pool', 'tvmonitor', 'concrete', 'box'

- DenseNet-161: 'muzzle', 'rock', 'bookcase', 'dog', 'airplane', 'footboard', 'seat', 'car', 'sea', 'screen', 'metal', 'leg', 'labyrinth', 'fence', 'food', 'ear', 'cat', 'bird', 'sky', 'body', 'roof', 'person', 'stove', 'sink', 'ceiling', 'skyscraper', 'grass', 'bus', 'road', 'house', 'pool table', 'ball', 'tree', 'cow', 'fountain', 'motorbike', 'bottle', 'desk', 'boat', 'tile', 'mountain', 'snow', 'field', 'bicycle', 'seat cushion', 'curtain', 'court', 'clouds', 'building', 'plate', 'leather', 'toilet', 'hair', 'head', 'blue-c', 'coach', 'shop window', 'bed', 'drawer', 'bridge', 'swimming pool', 'washer', 'wing', 'hand', 'tent', 'stairway', 'truck', 'barrel', 'aircraft carrier', 'case', 'chair', 'wheel', 'slot machine', 'ceramic', 'wood', 'bathtub', 'horse', 'water', 'stern', 'book', 'concrete', 'red-c', 'carpet', 'train', 'torso', 'pot', 'carousel', 'painting', 'railing', 'table', 'cabinet', 'yellow-c', 'signboard', 'sheep', 'white-c',

Class	Concept
applauding	[airplane, person, hair, dog, seat, sea, cat, screen, car, bus]
blowing_bubbles	[footboard, airplane, bus, hair, building, person, head, horse, dog, tent]
brushing_teeth	[person, airplane, bus, seat, cat, hair, head, house, food, dog]
cleaning_the_floor	[dog, airplane, pool table, bird, house, book, sea, seat, carpet, screen]
climbing	[airplane, mountain, screen, rock, dog, house, person, roof, wardrobe, carpet]
cooking	[airplane, metal, food, rock, motorbike, person, case, bottle, plate, sink]
cutting_trees	[airplane, tree, mountain, dog, seat cushion, person, field, rock, bird, carpet]
cutting_vegetables	[airplane, food, case, metal, hair, person, rock, tree, flower, seat]
drinking	[airplane, seat, person, hair, dog, food, sea, bus, house, bottle]
feeding_a_horse	[airplane, horse, house, dog, mountain, carpet, field, grass, person, building]
fishing	[mountain, airplane, sea, water, skyscraper, dog, muzzle, wardrobe, car, person]
fixing_a_bike	[bicycle, airplane, motorbike, mountain, wheel, person, train, building, house, bird]
fixing_a_car	[airplane, car, motorbike, person, dog, building, mountain, road, train, house]
gardening	[airplane, tree, plant, mountain, dog, person, grass, snow, sea, house]
holding_an_umbrella	[tent, roof, airplane, mountain, building, car, dog, cow, person, table]
jumping	[airplane, dog, mountain, cat, house, sea, food, person, wardrobe, tree]
looking_through_a_microscope	[airplane, seat, hair, plate, screen, metal, person, muzzle, house, table]
looking_through_a_telescope	[airplane, screen, body, head, mountain, bicycle, sea, dog, pool table, person]
phoning	[airplane, hair, person, screen, building, seat, house, sea, dog, cat]
playing_guitar	[horse, sheep, airplane, train, bookcase, car, person, seat, screen, mountain]
playing_violin	[dog, airplane, bottle, horse, food, person, bird, train, body, hair]
pouring_liquid	[airplane, person, metal, hair, dog, seat, plate, food, bottle, table]
pushing_a_cart	[airplane, carpet, building, dog, cradle, bird, screen, bus, road, person]
reading	[airplane, hair, person, dog, screen, seat, sea, sofa, bird, bus]
riding_a_bike	[bicycle, airplane, motorbike, mountain, building, wheel, screen, dog, house, tree]
riding_a_horse	[airplane, horse, house, grass, field, mountain, dog, carpet, screen, motorbike]
rowing_a_boat	[airplane, water, mountain, dog, motorbike, pool table, sea, skyscraper, book, horse]
running	[airplane, dog, road, mountain, sea, building, person, swimming pool, food, bus]
shooting_an_arrow	[airplane, bird, metal, muzzle, mountain, court, head, bicycle, dog, house]
smoking	[airplane, hair, person, building, seat, sea, dog, screen, head, house]
taking_photos	[airplane, screen, house, dog, person, building, seat, sea, mountain, food]
texting_message	[airplane, hair, person, seat, sea, screen, house, dog, mountain, cat]
throwing_frisby	[dog, airplane, grass, sea, skyscraper, mountain, field, bus, swimming pool, person]
using_a_computer	[screen, airplane, seat, hair, table, bookcase, sea, lamp, footboard, mountain]

Class	Concept
walking_the_dog	[dog, sea, airplane, mountain, person, building, carpet, bus, screen, road]
washing_dishes	[airplane, sink, rock, metal, seat, plate, house, tree, lamp, bus]
watching_TV	[airplane, screen, seat, tree, lamp, sea, bed, sofa, carpet, hair]
waving_hands	[airplane, person, dog, hair, seat, cat, sea, mountain, house, bus]
writing_on_a_board	[airplane, seat, hair, person, screen, dog, house, cat, tile, mountain]
writing_on_a_book	[airplane, hair, person, seat, table, screen, bird, mountain, dog, cat]

Table B.7: Top 10 Local Concepts - Action40/DenseNet-161

'pink-c', 'chandelier', 'orange-c', 'sand', 'purple-c', 'plant', 'pane', 'waterfall', 'canopy', 'windowpane', 'lid', 'shelf', 'arcade machine', 'lamp', 'wardrobe', 'frame', 'black-c', 'laptop', 'refrigerator', 'cradle', 'sidewalk', 'sofa', 'fabric', 'balcony', 'apparel', 'shade', 'flower', 'brick', 'fireplace', 'arm', 'track', 'box'

B.2 Networks hyper-parameters and training

B.2.1 Resnet-152

ResNet-152 [51] belongs to the *ResNet* family of architectures, which introduced the concept of residual learning to tackle the challenges associated with training very deep neural networks. A significant innovation of *ResNet* is the implementation of residual blocks, which enhance the flow of gradients during backpropagation. This is achieved by incorporating shortcut connections that bypass one or more layers, enabling the network to learn residual functions rather than unreferenced mappings. With its 152 layers, *ResNet-152* is among the deeper models in the *ResNet* family. It has proven to achieve high accuracy on tasks such as *ImageNet* classification while maintaining lower complexity compared to other deep networks.

In our experiments, we substituted the fully connected layer for a *SVM* with a linear kernel for the transfer-learning process. The concepts were extracted from the last convolutional layer, named "layer_4".

Class	Concept
applauding	[seat, hair, rock, dog, airplane, bus, train]
blowing_bubbles	[footboard, horse, tent, bird, coach, road, bottle, bus, head]
brushing_teeth	[cat, car, bottle, screen, head, snow, lamp, horse, bus, food]
cleaning_the_floor	[dog, book, pool table, bird, house, airplane, body, sink, rock]
climbing	[rock, seat, brick, mountain, tree, concrete, metal, ceiling, wardrobe, house]
cooking	[food, metal, airplane, slot machine, body, bed, table]
cutting_trees	[bird, tree, building, head, roof, body, sink, dog]
cutting_vegetables	[arm, food, pool table, metal, muzzle, person]
drinking	[horse, arm, sidewalk, bottle, head, house, bed, sink, washer]
feeding_a_horse	[horse, dog, bus, carpet, hair, fence, cow, cat]
fishing	[court, muzzle, water, car, sea, bus, motorbike, skyscraper, airplane]
fixing_a_bike	[bicycle, wheel, person, boat, motorbike, head, arm]
fixing_a_car	[car, airplane, train, snow, bed]
gardening	[plant, grass, tree, pot, flower, lid, food, windowpane, track]
holding_an_umbrella	[tent, cow, roof, car, mountain, labyrinth, bridge, table]
jumping	[airplane, leg, cat, grass, carpet, leather, lid, dog]
looking_through_a_microscope	[plate, metal, muzzle, hair, cat, house, bird, torso, airplane]
looking_through_a_telescope	[head, bicycle, pool table, body, cat, sea, field, horse, airplane]
phoning	[tent, screen, ceiling, house, sea, pot, drawer, hair]
playing_guitar	[sheep, horse, bookcase, train, red-c, chair, car, wing, dog, food]
playing_violin	[bottle, food, dog, bird, chair, body, train]
pouring_liquid	[water, plate, bottle, body, toilet, building, motorbike, stove]
pushing_a_cart	[pot, bird, road, railing, screen, cow, cradle, muzzle, building]
reading	[book, food, bookcase, sheep, body, pot, chair, bird, carpet, dog]
riding_a_bike	[bicycle, motorbike, person]
riding_a_horse	[bicycle, painting, court, horse, carpet, food, sand, screen, bus, body]
rowing_a_boat	[pool table, motorbike, water, dog, sofa, book, red-c]
running	[bicycle, horse, food, road, sheep, leather, house, body]
shooting_an_arrow	[muzzle, bird, head, metal, bicycle, airplane, rock, tent, bed, torso]
smoking	[head, house, coach, bus, hair, food, person, train, case, sea]
taking_photos	[sink, house, screen, dog, sea, building, torso, body, plant, bus]
texting_message	[house, arcade machine, torso, person, body, dog, tent, cat, airplane, bookcase]
throwing_frisby	[dog, skyscraper, bus, field, house, boat, swimming pool, painting, bridge, bird]

Class	Concept
using_a_computer	[screen, bird, mountain, airplane, skyscraper]
walking_the_dog	[dog, snow, food, bus, bottle, person, road]
washing_dishes	[fountain, sink, labyrinth, toilet, bus, rock, stove, motorbike, dog]
watching_TV	[screen, washer, airplane, seat cushion, sea, dog, signboard]
waving_hands	[stove, hair, field, person, hand, horse, metal, plant]
writing_on_a_board	[airplane, tile, hair, tree, screen, bird, horse, truck, brick]
writing_on_a_book	[box, sheep, tent, mountain, skyscraper, cat, sea, toilet, stove, screen]

Table B.8: Top 10 Global Concepts - Action40/DenseNet-161

	Train	Test
Airplane	5000	1000
Automobile	5000	1000
Bird	5000	1000
Cat	5000	1000
Deer	5000	1000
Dog	5000	1000
Frog	5000	1000
Horse	5000	1000
Ship	5000	1000
Truck	5000	1000

Table B.9: CIFAR-10 Dataset Distribution

Class	Concepts
airplane	[seat cushion, stove, airplane, boat, plant, sea, person, smoke, body, sky]
automobile	[seat cushion, stove, person, plant, car, dog, body, boat, bus, house]
bird	[seat cushion, bird, plant, smoke, dog, body, stove, food, sea, rock]
cat	[seat cushion, dog, plant, cat, smoke, body, stove, rock, bird, hand]
deer	[seat cushion, bird, plant, grass, dog, field, smoke, rock, water, sea]
dog	[seat cushion, dog, plant, stove, cat, body, water, smoke, bed, horse]
frog	[seat cushion, plant, food, smoke, bird, dog, rock, hand, sea, cat]
horse	[seat cushion, plant, horse, person, dog, grass, water, stove, bird, rock]
ship	[seat cushion, boat, stove, airplane, sea, plant, water, person, body, pool table]
truck	[seat cushion, person, dog, stove, plant, boat, house, motorbike, airplane, bus]

Table B.10: Top 10 Local Concepts - CIFAR-10/ResNet-152

Class	Concept
airplane	[airplane, seat, muzzle, road, horse, building, chandelier]
automobile	[table, seat, car, body, shelf, bus, bicycle, mountain]
bird	[metal, food, bird, cow, dog, road, horse, footboard]
cat	[cat, head, dog, table, horse, torso, food, washer]
deer	[bird, head, pot, food, dog, cat, chair, footboard, signboard]
dog	[dog, muzzle, skyscraper, washer, torso, head, water, hair]
frog	[food, cat, hand, plant, bird, leg, bus, dog, skyscraper, labyrinth]
horse	[sink, horse, fence, cat, dog]
ship	[boat, body, sea, fireplace, water, shelf, cat]
truck	[train, dog, body, motorbike, bicycle, food, metal, cat]

Table B.11: Top 10 Global Concepts - CIFAR-10/ResNet-152

Class	Concept
airplane	[dog, pool table, cabinet, ceramic, wing, water, tree, airplane, sky, sea]
automobile	[pool table, dog, cabinet, ceramic, car, wing, train, tree, sea, motorbike]
bird	[pool table, cabinet, dog, pot, bird, ceramic, grass, wing, fur, field]
cat	[pool table, dog, cabinet, pot, ceramic, cat, wing, fur, person, headboard]
deer	[pool table, cabinet, dog, grass, field, pot, tree, fur, person, plant]
dog	[pool table, dog, cabinet, ceramic, pot, cat, person, field, wing, sheep]
frog	[pool table, dog, cabinet, pot, food, rock, fur, ceramic, grass, plant]
horse	[dog, cabinet, pool table, ceramic, tree, grass, horse, person, field, water]
ship	[dog, pool table, cabinet, ceramic, water, boat, sea, wing, sky, pot]
truck	[cabinet, pool table, dog, ceramic, train, tree, wing, car, coach, body]

Table B.12: Top 10 Local Concepts - CIFAR-10/ResNet-50

Class	Concept
airplane	[body, sky, airplane, smoke, stern, person, horse, pool table]
automobile	[tree, car, sea, screen, stove, bird, motorbike]
bird	[grass, fence, bird, headboard, seat, motorbike, cat]
cat	[cat, pool table, cow, horse, bird, bus]
deer	[field, cow, bird, dog, toilet, horse, plant, pool table, fireplace]
dog	[dog, house, bed, cat, crosswalk]
frog	[drawer, dog, cat, sink, head, mountain, bathtub, carpet, pottedplant, food]
horse	[person, house, food, chair, cat, washer, sheep, seat]
ship	[water, sea, boat, airplane, building, carousel]
truck	[dog, tree, bus, wheel, red-c, car, road, food, coach]

Table B.13: Top 10 Global Concepts - CIFAR-10/ResNet-50

Class	Concept
airplane	[dog, airplane, muzzle, bird, body, car, house, sea, grass, rock]
automobile	[dog, muzzle, airplane, car, pool table, bird, bus, house, fountain, tree]
bird	[dog, muzzle, airplane, bird, rock, tree, body, food, person, arcade machine]
cat	[muzzle, dog, airplane, cat, rock, bird, person, food, screen, bus]
deer	[dog, muzzle, airplane, tree, rock, field, bird, mountain, grass, seat cushion]
dog	[dog, muzzle, airplane, bird, rock, person, cat, screen, food, house]
frog	[muzzle, dog, airplane, bird, rock, food, person, tree, field, metal]
horse	[muzzle, dog, airplane, horse, grass, field, bicycle, tree, painting, house]
ship	[muzzle, dog, airplane, sea, water, fountain, pool table, bottle, bird, house]
truck	[dog, muzzle, airplane, house, bus, train, pool table, car, tree, cat]

Table B.14: Top 10 Local Concepts - CIFAR-10/DenseNet-161

Class	Concept
airplane	[airplane, mountain, horse, sky, dog]
automobile	[car, cabinet, screen, fountain, case, pool table]
bird	[food, body, bird, sink, fence, book, screen, person]
cat	[sea, tent, roof, cat, seat, sheep, ear, hand, lamp]
deer	[person, case, rock, bottle, bicycle, seat cushion, table, house, sheep, ceiling]
dog	[dog, pot, food, ceiling, horse, chair, building, cradle, house]
frog	[plant, washer, cat, house, food, hair, person, bus, sink, drawer]
horse	[house, motorbike, chair, car, horse, sea, metal]
ship	[water, sea, motorbike, screen, boat, food, airplane, coach, desk]
truck	[cat, pane, roof, coach, dog, airplane, tree, toilet, house]

Table B.15: Top 10 Global Concepts - CIFAR-10/DenseNet-161

B.2.2 Resnet-50

ResNet-50 [51] also belongs to the *ResNet* family of architectures, but it is a smaller variant with 50 layers that balances depth and computational efficiency, making it suitable for a wide range of applications, including medical image analysis and face recognition systems.

In our experiments, we substituted the fully connected layer for a *SVM* with a linear kernel for the transfer-learning process. The concepts were extracted from the last convolutional layer, named "layer_4".

B.2.3 Densenet-161

DenseNet-161 [55] is part of the *DenseNet* architecture family, characterised by its densely connected convolutional networks. Unlike traditional convolutional networks, where each layer connects only to its subsequent layer, DenseNets establish direct connections from each layer to all subsequent layers. Such connectivity improves information flow between layers, mitigates the vanishing gradient problem, enhances feature reuse, and reduces the number of parameters needed compared to traditional architectures.

In our experiments, we substituted the fully connected layer for a *SVM* with a linear kernel for the transfer-learning process. The concepts were extracted from the last convolutional layer, named "features".

Appendix C

Presentations on Work from this Thesis

- CRT-AI Student Seminar (2021 & 2022)
- Thesis in 3 - Research Day at Dublin City University (2023)
- Lecture for a master's class at University College Dublin (2022 & 2023)
- Imperial College SPIKE group meeting (2023)
- Research Day at University of Amsterdam (2024)

Bibliography

[1] OpenAI Josh Achiam et al. "GPT-4 Technical Report". In: (2023). URL: https://api.semanticscholar.org/CorpusID:257532815.

[2] Julius Adebayo et al. *Debugging tests for model explanations*. Vancouver, BC, Canada, 2020.

[3] Hamed Habibi Aghdam, Elnaz Jahani Heravi, et al. "Guide to convolutional neural networks". In: *New York, NY: Springer* 10.978-973 (2017), p. 51.

[4] Sajid Ali et al. "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence". In: *Information Fusion* 99 (2023), p. 101805. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2023.101805. URL: https://www.sciencedirect.com/science/article/pii/S1566253523001148.

[5] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information fusion* 58 (2020), pp. 82–115.

[6] Yaniv Aspis et al. "Embed2Rule Scalable Neuro-Symbolic Learning via Latent Space Weak-Labelling". In: *Neural-Symbolic Learning and Reasoning*. Ed. by Tarek R. Besold et al. Cham: Springer Nature Switzerland, 2024, pp. 195–218. ISBN: 978-3-031-71167-1.

[7] Sören Auer et al. "Dbpedia: A nucleus for a web of open data". In: *international semantic web conference*. Springer. 2007, pp. 722–735.

[8] M Gethsiyal Augasta and Thangairulappan Kathirvalavakumar. "Reverse engineering the neural networks for rule extraction in classification problems". In: *Neural processing letters* 35 (2012), pp. 131–150.

[9] AN Averkin and SA Yarushev. "Review of research in the field of developing methods to extract rules from artificial neural networks". In: *Journal of Computer and Systems Sciences International* 60 (2021), pp. 966–980.

[10] Reza Azad et al. "Laplacian-former: Overcoming the limitations of vision transformers in local texture detection". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2023, pp. 736–746.

[11] Eric B. Bartlett. "Self determination of input variable importance using neural networks". In: *Neural, Parallel & Scientific Computations* 2 (Mar. 1994), pp. 103–114.

[12] Sai Balaji. "Binary Image classifier CNN using TensorFlow". In: (2020). Online; accessed 15-October-2024. URL: https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697.

[13] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. URL: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[14] David Bau et al. "Network Dissection: Quantifying Interpretability of Deep Visual Representations". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3319–3327. DOI: 10.1109/CVPR.2017.354.

[15] David Bau et al. "Understanding the role of individual units in a deep neural network". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30071–30078. DOI: 10.1073/pnas.1907375117. eprint: https://www.

pnas.org/doi/pdf/10.1073/pnas.1907375117. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1907375117.

[16] Kexin Gu Baugh, Nuri Cingillioglu, and Alessandra Russo. *Neuro-symbolic Rule Learning in Real-world Classification Tasks*. 2023. arXiv: 2303.16674 [cs.LG]. URL: https://arxiv.org/abs/2303.16674.

[17] Kurt Bollacker et al. "Freebase: a collaboratively created graph database for structuring human knowledge". In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 2008, pp. 1247–1250.

[18] Guido Bologna. "A simple convolutional neural network with rule extraction". In: *Applied Sciences* 9.12 (2019), p. 2411.

[19] Guido Bologna and Silvio Fossati. "A two-step rule-extraction technique for a cnn". In: *Electronics* 9.6 (2020), p. 990.

[20] Manomita Chakraborty, Saroj Kumar Biswas, and Biswajit Purkayastha. "Rule extraction from neural network trained using deep belief network and back propagation". In: *Knowledge and Information Systems* 62.9 (2020), pp. 3753–3781.

[21] Yupeng Chang et al. "A survey on evaluation of large language models". In: *ACM Transactions on Intelligent Systems and Technology* 15.3 (2024), pp. 1–45.

[22] Xu Cheng et al. "Explaining Knowledge Distillation by Quantifying the Knowledge". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 12922–12932. DOI: 10.1109/CVPR42600.2020.01294.

[23] Shivani Choudhary et al. "A survey of knowledge graph embedding and their applications". In: *arXiv preprint arXiv:2107.07842* (2021).

[24] Michele Collevati, Thomas Eiter, and Nelson Higuera. "Leveraging Neurosymbolic AI for Slice Discovery". In: *Neural-Symbolic Learning and Reasoning*. Ed. by Tarek R. Besold et al. Cham: Springer Nature Switzerland, 2024, pp. 403–418. ISBN: 978-3-031-71167-1.

[25] Domenico Corapi, Alessandra Russo, and Emil Lupu. "Inductive Logic Programming in Answer Set Programming". In: *Inductive Logic Programming*. Ed. by Stephen H. Muggleton, Alireza Tamaddoni-Nezhad, and Francesca A. Lisi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 91–97. ISBN: 978-3-642-31951-8.

[26] Mark Craven and Jude Shavlik. "Extracting Tree-Structured Representations of Trained Networks". In: *Advances in Neural Information Processing Systems*. Ed. by D. Touretzky, M.C. Mozer, and M. Hasselmo. Vol. 8. MIT Press, 1995. URL: https://proceedings.neurips.cc/paper_files/paper/1995/file/45f31d16b1058d586fc3be7207b58053-Paper.pdf.

[27] Andrew Cropper and Sebastijan Dumančić. "Inductive logic programming at 30: a new introduction". In: *Journal of Artificial Intelligence Research* 74 (2022), pp. 765–850.

[28] Yan Cui et al. "CEIR: Concept-based Explainable Image Representation Learning". In: (2024). URL: https://openreview.net/forum?id=zhJDD85QHD.

[29] Daniel Cunnington et al. "Ffnsl: Feed-forward neural-symbolic learner". In: *Machine Learning* 112.2 (2023), pp. 515–569.

[30] Daniel Cunnington et al. "Symbolic Learning for Material Discovery". In: *arXiv preprint arXiv:2312.11487* (2023).

[31] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[32] Shumin Deng et al. “Knowledge-driven stock trend prediction and explanation via temporal convolutional network”. In: *Companion proceedings of the 2019 world wide web conference*. 2019, pp. 678–685.

[33] Eric Ferreira Dos Santos, Danilo Silva de Carvalho, and Jonice Oliveira. “Pattern Identification of Bot Messages for Media Literacy”. In: *Proceedings of the Brazilian Symposium on Multimedia and the Web*. 2021, pp. 121–128.

[34] Arthur Drozdov et al. “Online symbolic learning of policies for explainable security”. In: *2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE. 2021, pp. 269–278.

[35] Darren Edge et al. *From Local to Global: A Graph RAG Approach to Query-Focused Summarization*. 2024. arXiv: 2404.16130 [cs.CL]. URL: https://arxiv.org/abs/2404.16130.

[36] Thomas Eiter, Giovambattista Ianni, and Thomas Krennwallner. “Answer Set Programming: A Primer”. In: *Reasoning Web. Semantic Technologies for Information Systems: 5th International Summer School 2009, Brixen-Bressanone, Italy, August 30 - September 4, 2009, Tutorial Lectures*. Ed. by Sergio Tessaris et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 40–110. ISBN: 978-3-642-03754-2. DOI: 10.1007/978-3-642-03754-2_2. URL: https://doi.org/10.1007/978-3-642-03754-2_2.

[37] Eric Ferreira dos Santos and Alessandra Mileo. “From Disentangled Representation to Concept Ranking: Interpreting Deep Representations in Image Classification Tasks”. In: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*. Springer. 2023, pp. 322–335.

[38] Claas Flint et al. "Systematic misestimation of machine learning performance in neuroimaging studies of depression". In: *Neuropsychopharmacology* 46.8 (2021), pp. 1510–1517.

[39] Johannes Fürnkranz. "Rule Learning". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 875–879. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_738. URL: https://doi.org/10.1007/978-0-387-30164-8_738.

[40] Artur d'Avila Garcez and Luis C Lamb. "Neurosymbolic AI: The 3 rd wave". In: *Artificial Intelligence Review* 56.11 (2023), pp. 12387–12406.

[41] Martin Gebser et al. "Clingo= ASP+ control: Preliminary report". In: *arXiv preprint arXiv:1405.3694* (2014).

[42] Michael Gelfond and Vladimir Lifschitz. "The Stable Model Semantics for Logic Programming". In: *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, USA, August 15-19, 1988 (2 Volumes)*. Ed. by Robert A. Kowalski and Kenneth A. Bowen. MIT Press, 1988, pp. 1070–1080.

[43] Leilani H. Gilpin et al. *Explaining Explanations: An Overview of Interpretability of Machine Learning*. 2018. arXiv: 1806.00069 [cs.AI].

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[45] Gudmund Grov et al. "On the Use of Neurosymbolic AI for Defending Against Cyber Attacks". In: *Neural-Symbolic Learning and Reasoning*. Ed. by Tarek R. Besold et al. Cham: Springer Nature Switzerland, 2024, pp. 119–140. ISBN: 978-3-031-71167-1.

[46] D. Gunning et al. "XAI-Explainable artificial intelligence". In: *Science Robotics* 4.37 (Dec. 2019). This is the author?s version of the work. It is posted here by permission of the AAAS for personal use, not for redistribution. The definitive version was published in Science Robotics on 4 (37) 18 December 2019,

DOI: 10.1126/scirobotics.aay7120., eaay7120. DOI: 10.1126/scirobotics.aay7120. URL: https://openaccess.city.ac.uk/id/eprint/23405/.

[47] Muhammad Usman Hadi et al. "Large Language Models: A Comprehensive Survey of its Applications, Challenges, Limitations, and Future Prospects". In: (2024). DOI: 10.36227/techrxiv.23589741.v7. URL: http://dx.doi.org/10.36227/techrxiv.23589741.v7.

[48] Tameru Hailesilassie. *Rule Extraction Algorithm for Deep Neural Networks: A Review*. 2016. arXiv: 1610.05267 [cs.CV]. URL: https://arxiv.org/abs/1610.05267.

[49] Tameru Hailesilassie. "Rule extraction algorithm for deep neural networks: A review". In: *arXiv preprint arXiv:1610.05267* (2016).

[50] Shibo Hao et al. *BertNet: Harvesting Knowledge Graphs with Arbitrary Relations from Pretrained Language Models*. Ed. by Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki. Toronto, Canada, July 2023. DOI: 10.18653/v1/2023.findings-acl.309. URL: https://aclanthology.org/2023.findings-acl.309/.

[51] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[52] Andreas Holzinger et al. "Explainable AI methods-a brief overview". In: *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*. Springer. 2022, pp. 13–38.

[53] Vitor A. C. Horta and Alessandra Mileo. "Towards Explaining Deep Neural Networks Through Graph Analysis". In: *Database and Expert Systems Applications*. Ed. by Gabriele Anderst-Kotsis et al. Cham: Springer International Publishing, 2019, pp. 155–165. ISBN: 978-3-030-27684-3.

[54] Zhiting Hu et al. "Harnessing Deep Neural Networks with Logic Rules". In: (Aug. 2016). Ed. by Katrin Erk and Noah A. Smith, pp. 2410–2420. DOI: 10.18653/v1/P16-1228. URL: https://aclanthology.org/P16-1228/.

[55] Gao Huang et al. *Densely Connected Convolutional Networks*. 2017. DOI: 10.1109/CVPR.2017.243.

[56] Rodrigo Toro Icarte et al. "How a general-purpose commonsense ontology can improve performance of learning-based image retrieval". In: IJCAI'17 (2017), pp. 1283–1289.

[57] Filip Ilievski, Pedro Szekely, and Bin Zhang. "Cskg: The commonsense knowledge graph". In: *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings 18*. Springer. 2021, pp. 680–696.

[58] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. *Understanding Convolutional Neural Networks for Text Classification*. 2018. URL: https://api.semanticscholar.org/CorpusID:52346770.

[59] Vidhya Kamakshi and Narayanan C Krishnan. "Explainable image classification: The journey so far and the road ahead". In: *AI* 4.3 (2023), pp. 620–651.

[60] Nikhil Kapila, Julian Glattki, and Tejas Rathi. *CNNtention: Can CNNs do better with Attention?* 2024. arXiv: 2412.11657 [cs.CV]. URL: https://arxiv.org/abs/2412.11657.

[61] Been Kim et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *International conference on machine learning*. PMLR. 2018, pp. 2668–2677.

[62] Jenia Kim, Henry Maathuis, and Danielle Sent. "Human-centered evaluation of explainable AI applications: a systematic review". In: *Frontiers in Artificial Intelligence* 7 (2024), p. 1456486.

[63] BA Kippfer. *Roget's 21st century thesaurus in dictionary form*. 1999.

[64] Pang Wei Koh et al. "Concept bottleneck models". In: *International conference on machine learning*. PMLR. 2020, pp. 5338–5348.

[65] Ranjay Krishna et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations". In: *International journal of computer vision* 123 (2017), pp. 32–73.

[66] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[67] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[68] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[69] Mark Law, Alessandra Russo, and Krysia Broda. "Inductive learning of answer set programs". In: *Logics in Artificial Intelligence: 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings 14*. Springer. 2014, pp. 311–325.

[70] Mark Law, Alessandra Russo, and Krysia Broda. "Inductive Learning of Answer Set Programs from Noisy Examples". In: *CoRR* abs/1808.08441 (2018). arXiv: 1808.08441. URL: http://arxiv.org/abs/1808.08441.

[71] Mark Law et al. "FastLAS: Scalable Inductive Logic Programming Incorporating Domain-Specific Optimisation Criteria". In: *AAAI Conference on Artificial Intelligence*. 2020. URL: https://api.semanticscholar.org/CorpusID:214191985.

[72] Mark Law et al. "Representing and Learning Grammars in Answer Set Programming". In: *Proceedings of the AAAI Conference on Artificial Intelligence*

33.01 (July 2019), pp. 2919–2928. DOI: 10.1609/aaai.v33i01.33012919. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4147.

[73] Freddy Lécué. "On the role of knowledge graphs in explainable AI." In: *Semantic Web* 11.1 (2020), pp. 41–51. DOI: 10.3233/SW-190374. URL: http://dblp.uni-trier.de/db/journals/semweb/semweb11.html#Lecue20.

[74] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[75] Jens Lehmann, Sebastian Bader, and Pascal Hitzler. "Extracting reduced logic programs from artificial neural networks". In: *Applied intelligence* 32 (2010), pp. 249–266.

[76] Vladimir Lifschitz. "What Is Answer Set Programming?" In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*. Ed. by Dieter Fox and Carla P. Gomes. AAAI Press, 2008, pp. 1594–1597. URL: http://www.aaai.org/Library/AAAI/2008/aaai08-270.php.

[77] Baoyuan Liu et al. "Sparse convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 806–814.

[78] Xiao Liu et al. "Learning disentangled representations in the imaging domain". In: *Medical Image Analysis* 80 (2022), p. 102516. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2022.102516. URL: https://www.sciencedirect.com/science/article/pii/S1361841522001633.

[79] Pedro Lopes et al. "XAI Systems Evaluation: A Review of Human and Computer-Centred Methods". In: *Applied Sciences* 12.19 (2022). ISSN: 2076-3417. DOI: 10.3390/app12199423. URL: https://www.mdpi.com/2076-3417/12/19/9423.

[80] Zhiying Lu et al. "Bridging the Gap Between Vision Transformers and Convolutional Neural Networks on Small Datasets". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=bfz-jhJ8wn.

[81] Scott M Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 4765–4774. URL: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[82] Nicolas Eric Maillot and Monique Thonnat. "Ontology based complex object recognition". In: *Image and Vision Computing* 26.1 (2008), pp. 102–113.

[83] B. Mak and R. W. Blanning. "An empirical measure of element contribution in neural networks". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 28.4 (Nov. 1998), pp. 561–564. ISSN: 1094-6977.

[84] Abhishek Mandal, Susan Leavy, and Suzanne Little. "Biased Attention: Do Vision Transformers Amplify Gender Bias More than Convolutional Neural Networks?" In: (2023). URL: https://papers.bmvc2023.org/0629.pdf.

[85] Andrei Margeloiu et al. "Do concept bottleneck models learn as intended?" In: *arXiv preprint arXiv:2105.04289* (2021).

[86] Sara El Mekkaoui, Loubna Benabbou, and Abdelaziz Berrado. *Rule-Extraction Methods From Feedforward Neural Networks: A Systematic Literature Review*. 2023. arXiv: 2312.12878 [cs.LG]. URL: https://arxiv.org/abs/2312.12878.

[87] Alessandra Mileo. "Towards a neuro-symbolic cycle for human-centered explainability". In: *Neurosymbolic Artificial Intelligence* (2024). URL: https://api.semanticscholar.org/CorpusID:272292970.

[88] Dang Minh et al. “Explainable artificial intelligence: a comprehensive review”. In: *Artificial Intelligence Review* (2021), pp. 1–66.

[89] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. “Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges”. In: *ECML PKDD 2020 Workshops*. Ed. by Irena Koprinska et al. Cham: Springer International Publishing, 2020, pp. 417–431. ISBN: 978-3-030-65965-3.

[90] Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. “Convolutional Neural Networks”. In: *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. Cham: Springer International Publishing, 2022, pp. 533–577. ISBN: 978-3-030-89010-0. DOI: 10.1007/978-3-030-89010-0_13. URL: https://doi.org/10.1007/978-3-030-89010-0_13.

[91] Stephen Muggleton. “Inverse entailment and Progol”. In: *New generation computing* 13 (1995), pp. 245–286.

[92] Meike Nauta et al. “From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI”. In: *ACM Comput. Surv.* 55.13s (July 2023). ISSN: 0360-0300. DOI: 10.1145/3583558. URL: https://doi.org/10.1145/3583558.

[93] Tuomas Oikarinen and Tsui-Wei Weng. “CLIP-Dissect: Automatic Description of Neuron Representations in Deep Vision Networks”. In: (2023). URL: https://openreview.net/forum?id=iPWiwWHc1V.

[94] Tuomas Oikarinen et al. “Label-free Concept Bottleneck Models”. In: (2023). URL: https://openreview.net/forum?id=FlCg47MNvBA.

[95] Patrick Perret. “Children’s inductive reasoning: Developmental and educational perspectives”. In: *Journal of Cognitive Education and Psychology* 14.3 (2015), pp. 389–408.

[96] Andrés Felipe Posada-Moreno et al. “Scale-preserving automatic concept extraction (SPACE)”. In: *Machine Learning* 112.11 (2023), pp. 4495–4525.

[97] J. Ross Quinlan. "Learning logical definitions from relations". In: *Machine learning* 5 (1990), pp. 239–266.

[98] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[99] Luc De Raedt. "Inductive Logic Programming". In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 529–537. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_396. URL: https://doi.org/10.1007/978-0-387-30164-8_396.

[100] Enayat Rajabi and Kobra Etminani. "Knowledge-graph-based explainable AI: A systematic review". In: *Journal of Information Science* 50.4 (2024), pp. 1019–1029. DOI: 10.1177/01655515221112844. eprint: https://doi.org/10.1177/01655515221112844. URL: https://doi.org/10.1177/01655515221112844.

[101] Enayat Rajabi and Somayeh Kafaie. "Knowledge Graphs and Explainable AI in Healthcare". In: *Information* 13.10 (2022). ISSN: 2078-2489. DOI: 10.3390/info13100459. URL: https://www.mdpi.com/2078-2489/13/10/459.

[102] Gabrielle Ras et al. "Explainable Deep Learning: A Field Guide for the Uninitiated". In: *J. Artif. Int. Res.* 73 (May 2022). ISSN: 1076-9757. DOI: 10.1613/jair.1.13200. URL: https://doi.org/10.1613/jair.1.13200.

[103] Scott Reed et al. "Learning Deep Representations of Fine-Grained Visual Descriptions". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, pp. 49–58. DOI: 10.1109/CVPR.2016.13. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.13.

[104] 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, San Francisco, CA, USA, August 13-17, 2016. 2016, pp. 1135–1144.

[105] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[106] Waddah Saeed and Christian Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities". In: *Knowledge-Based Systems* 263 (2023), p. 110273. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2023.110273. URL: https://www.sciencedirect.com/science/article/pii/S0950705123000230.

[107] Maarten Sap et al. "Atomic: An atlas of machine commonsense for if-then reasoning". In: *Proceedings of the AAAI conference on artificial intelligence*. 2019, pp. 3027–3035.

[108] Simon Schramm, Christoph Wehner, and Ute Schmid. "Comprehensible Artificial Intelligence on Knowledge Graphs: A survey". In: *Journal of Web Semantics* 79 (2023), p. 100806. ISSN: 1570-8268. DOI: https://doi.org/10.1016/j.websem.2023.100806. URL: https://www.sciencedirect.com/science/article/pii/S1570826823000355.

[109] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[110] Prithviraj Sen et al. "Neuro-Symbolic Inductive Logic Programming with Logical Neural Networks". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.8 (June 2022), pp. 8212–8219. DOI: 10.1609/aaai.v36i8.20795. URL: https://ojs.aaai.org/index.php/AAAI/article/view/20795.

[111] Saleh Shahinfar, Paul Meek, and Greg Falzon. ""How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring". In: *Ecological Informatics* 57 (2020), p. 101085. ISSN: 1574-9541. DOI: https://doi.org/10.1016/j.ecoinf.2020.101085. URL: https://www.sciencedirect.com/science/article/pii/S1574954120300352.

[112] Zohreh Shams et al. "REM: An Integrative Rule Extraction Methodology for Explainable Data Analysis in Healthcare". In: *medRxiv* (2021). DOI: 10.1101/2021.01.25.21250459. eprint: https://www.medrxiv.org/content/early/2021/04/28/2021.01.25.21250459.full.pdf. URL: https://www.medrxiv.org/content/early/2021/04/28/2021.01.25.21250459.

[113] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556.

[114] Margarita Sordo and Qing Zeng. "On Sample Size and Classification Accuracy: A Performance Comparison". In: *Biological and Medical Data Analysis*. Ed. by José Luís Oliveira et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 193–201. ISBN: 978-3-540-31658-9.

[115] Robyn Speer, Joshua Chin, and Catherine Havasi. "Conceptnet 5.5: An open multilingual graph of general knowledge". In: *Proceedings of the AAAI conference on artificial intelligence*. 2017.

[116] Jost Tobias Springenberg et al. "Striving for Simplicity: The All Convolutional Net". In: *CoRR* abs/1412.6806 (2014). URL: https://api.semanticscholar.org/CorpusID:12998557.

[117] Haixia Sun et al. "Medical knowledge graph to enhance fraud, waste, and abuse detection on claim data: Model development and performance evaluation". In: *JMIR Medical Informatics* 8.7 (2020), e17653.

[118] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[119] Ilaria Tiddi and Stefan Schlobach. "Knowledge graphs as tools for explainable machine learning: A survey". In: *Artificial Intelligence* 302 (2022), p. 103627. ISSN: 0004-3702. URL: https://www.sciencedirect.com/science/article/pii/S0004370221001788.

[120] Erico Tjoa and Cuntai Guan. "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.11 (2021), pp. 4793–4813. DOI: 10.1109/TNNLS.2020.3027314.

[121] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[122] Geoffrey G. Towell and Jude W. Shavlik. "Extracting Refined Rules from Knowledge-Based Neural Networks". In: *Mach. Learn.* 13.1 (Oct. 1993), pp. 71–101. ISSN: 0885-6125.

[123] Joseph Townsend, Thomas Chaton, and João M. Monteiro. "Extracting Relational Explanations From Deep Neural Networks: A Survey From a Neural-Symbolic Perspective". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.9 (2020), pp. 3456–3470. DOI: 10.1109/TNNLS.2019.2944672.

[124] Stefanos Tsimenidis. "Limitations of Deep Neural Networks: a discussion of G. Marcus' critical appraisal of deep learning". In: *arXiv preprint arXiv:2012.15754* (2020).

[125] Richard E. Turner. *An Introduction to Transformers*. 2024. arXiv: 2304.10557 [cs.LG]. URL: https://arxiv.org/abs/2304.10557.

[126] Danding Wang et al. "Designing theory-driven user-centric explainable AI". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–15.

[127] Ya Wang and Adrian Paschke. "Extracting Interpretable Hierarchical Rules from Deep Neural Networks' Latent Space". In: *Rules and Reasoning*. Ed. by Anna Fensel et al. Cham: Springer Nature Switzerland, 2023, pp. 238–253. ISBN: 978-3-031-45072-3.

[128] Sarah Wiegreffe and Yuval Pinter. "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 11–20. DOI: 10.18653/v1/D19-1002. URL: https://aclanthology.org/D19-1002/.

[129] Janusz Wojtusiak. "Rule Learning". In: *Encyclopedia of the Sciences of Learning*. Ed. by Norbert M. Seel. Boston, MA: Springer US, 2012, pp. 2909–2911. ISBN: 978-1-4419-1428-6. DOI: 10.1007/978-1-4419-1428-6_1347. URL: https://doi.org/10.1007/978-1-4419-1428-6_1347.

[130] Zhun Yang, Adam Ishay, and Joohyung Lee. *NeurASP: embracing neural networks into answer set programming*. Yokohama, Yokohama, Japan, 2021.

[131] Bangpeng Yao et al. "Human action recognition by learning bases of action attributes and parts". In: *2011 International conference on computer vision*. IEEE. 2011, pp. 1331–1338.

[132] Jay Yu, Kevin McCluskey, and Saikat Mukherjee. "Tax knowledge graph for a smarter and more personalized turbotax". In: *arXiv preprint arXiv:2009.06103* (2020).

[133] Mert Yuksekgonul, Maggie Wang, and James Zou. *Post-hoc Concept Bottleneck Models*. 2023. URL: https://openreview.net/forum?id=nA5AZ8CEyow.

[134] Mohammad Nokhbeh Zaeem and Majid Komeili. "Cause and Effect: Concept-based Explanation of Neural Networks". In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2021, pp. 2730–2736. DOI: 10.1109/SMC52423.2021.9658985.

[135] Mateo Espinosa Zarlenga, Zohreh Shams, and Mateja Jamnik. *Efficient Decompositional Rule Extraction for Deep Neural Networks*. 2021. arXiv: 2111.12628 [cs.LG]. URL: https://arxiv.org/abs/2111.12628.

[136] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[137] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8827–8836.

[138] Quanshi Zhang et al. "Interpreting cnns via decision trees". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6261–6270.

[139] Zhengyan Zhang et al. "ERNIE: Enhanced Language Representation with Informative Entities". In: (July 2019). Ed. by Anna Korhonen, David Traum, and Lluís Màrquez, pp. 1441–1451. DOI: 10.18653/v1/P19-1139. URL: https://aclanthology.org/P19-1139/.

[140] Menghua Zheng et al. "Attention-based CNNs for Image Classification: A Survey". In: *Journal of Physics: Conference Series* 2171.1 (Jan. 2022), p. 012068. DOI: 10.1088/1742-6596/2171/1/012068. URL: https://dx.doi.org/10.1088/1742-6596/2171/1/012068.

[141] Bolei Zhou et al. "Interpreting deep visual representations via network dissection". In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2131–2145.

[142] Qingguo Zhou et al. “Chapter 6 - Deep learning and visual perception”. In: *Theories and Practices of Self-Driving Vehicles*. Ed. by Qingguo Zhou et al. Elsevier, 2022, pp. 177–216. ISBN: 978-0-323-99448-4. DOI: https://doi.org/10.1016/B978-0-323-99448-4.00006-0. URL: https://www.sciencedirect.com/science/article/pii/B9780323994484000060.

[143] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. “DeepRED – Rule Extraction from Deep Neural Networks”. In: *Discovery Science*. Ed. by Toon Calders, Michelangelo Ceci, and Donato Malerba. Cham: Springer International Publishing, 2016, pp. 457–473. ISBN: 978-3-319-46307-0.

[144] Xiaohan Zou. “A survey on application of knowledge graph”. In: *Journal of Physics: Conference Series*. IOP Publishing. 2020, p. 012016.