Relational Explanations for Visual Domains: A Neural-symbolic Approach Combining ILP and CNNs

Johannes Rabold

Faculty for Information Systems and
Applied Computer Sciences

University of Bamberg

Bamberg 2024

Dissertation zur Erlangung des akademischen Grades doctor rerum naturalium (Dr. rer. nat.) der Fakultät Wirtschaftsinformatik und Angewandte Informatik der Otto-Friedrich-Universität Bamberg, vorgelegt von Johannes Rabold im Februar 2024.

Tag der mündlichen Prüfung: 06.06.2024

Erstgutachterin: Prof. Dr. Ute Schmid

Zweitgutachter: Prof. Dr. Diedrich Wolter

Mitglied der Promotionskommission: Prof. Dr. Daniela Nicklas

URN: urn:nbn:de:bvb:473-irb-1042696

DOI: https://doi.org/10.20378/irb-104269

Abstract

With powerful machine learning methods leaving the lab, the need for transparency in automatic decision processes becomes apparent. Only if humans have the ability to scrutinize how a model behaves and what the rationale behind a decision was, they will gain trust in a system. However, current neural network architectures are mainly black-box, so it is not easily possible to comprehend model decisions. Symbolic machine learning approaches that are inherently interpretable already exist for a long time in artificial intelligence research. However, most of these are not actually suitable for real world applications, since they lack the ability to work with raw (image) data, and thus, an accuracy-interpretability trade-off emerges.

The research branch of eXplainable Artificial Intelligence (XAI) advises interpretable surrogate models as a possible solution: Instead of abandoning a powerful model, it is kept as is and an additional interpretable model is generated, mimicking the behavior of the black-box. The gold standard for explaining image processing approaches like convolutional neural networks are visual attribution methods. For an image instance, methods like LIME or Grad-CAM output heatmaps, indicating regions that were influential (in a positive or negative way) for a particular model decision. These methods can give a first idea, what constituents in an image are important, and can pinpoint flaws in the trained model that can stem from e.g. bias in the training data. However, they lack expressiveness and can obscure the importance of e.g. relations that hold between image parts. This becomes particularly important for relational domains, where classification depends not only on the presence of image parts, but also their (spatial) constellation.

This thesis will describe and analyze methods of generating expressive symbolic relational explanations in the form of first-order logic rules. These rules inherently highlight the importance of not only visual concepts, but also of the relations between them. They also can be easily interpreted by humans and can even be converted to natural language in a straight-forward fashion. This work gives approaches for generating explanations in a variety of problem settings: Not only is it possible to explain the decision for a single image, it is also possible to explain the model as a whole. Also, when access to model parameters is given, explanations can benefit from that.

In an attempt to holistically approach relational explanation generation, this work will look at methods on how to be as close to the behavior of the original model as possible and also on how to quantify this "fidelity". Additionally, this work will revisit the generated explanations and ask, what types of explanations are particularly useful for humans.

Zusammenfassung (German Summary)

Titel: Relationale Erklärungen für Visuelle Domänen: Ein Neuro-symbolischer Ansatz zur Verknüpfung von ILP und CNNs

Mit dem Aufkommen von ausgereiften, leistungsstarken Machine-Learning-Methoden wird auch der Bedarf an Transparenz in automatischen Entscheidungssystemen deutlich. Nur wenn Menschen die Möglichkeit besitzen, zu hinterfragen wie sich ein Modell verhält, werden Sie auch Vertrauen in ein System aufbauen. Viele derzeitige neuronale Netzwerk-Architekturen haben jedoch überwiegend Black-Box-Charakter, welcher es nicht leicht macht, Modell-Entscheidungen nachzuvollziehen. Inhärent interpretierbare, symbolische Machine-Learning-Ansätze existieren schon lange Zeit in der KI-Forschung. Jedoch sind die meisten nicht gut geeignet für Anwendungen in der echten Welt, da sie nicht mit Roh(-Bild)-Daten umgehen können und damit ein Accuracy-Interpretability-Trade-Off einhergeht.

Das Forschungsfeld der Erklärbaren Künstlichen Intelligenz (XAI) bietet interpretierbare Stellvertreter-Modelle als mögliche Lösung: Anstatt der Abkehr von einem leistungsstarken Modell, werden dieses unverändert behalten und es wird ein zusätzliches, interpretierbares Modell generiert, welches das Verhalten der Black-Box imitiert. Der Gold-Standard um bildverarbeitende Methoden wie Convolutional Neural Networks zu erklären, sind visuelle Attributions-Verfahren. Methoden wie LIME oder Grad-CAM geben für eine Bild-Instanz Heatmaps aus, welche auf Regionen hinweisen, die (positiv oder negativ) einflussreich für eine gegebene Modell-Entscheidung waren. Diese Methoden können einen ersten Eindruck vermitteln, welche Bestandteile in einem Bild wichtig sind und können auf Defizite im trainierten Modell hinweisen, welche zum Beispiel von Bias in den Trainingsdaten stammen. Diesen Methoden fehlt es allerdings an Aussagekraft und sie können die Wichtigkeit von z.B. Relationen zwischen den Bild-Elementen verschleiern. Dies wird besonders in relationalen Domänen wichtig, bei denen die Klassifikation nicht nur von der Anwesenheit von Bild-Elementen abhängt, sondern auch von deren (räumlichen) Konstellation.

Diese Doktorarbeit wird Methoden bechreiben und analysieren, um aussagekräftige, symbolische und relationale Erklärungen in Form von prädikatenlogischen Regeln zu generieren. Diese Regeln heben von Natur aus nicht nur die Wichtigkeit von visuellen Konzepten, sondern auch deren Relationen untereinander hervor. Sie können ebenso einfach von Menschen interpretiert werden und können sogar ohne große Zwischenschritte in natürliche Sprache umgewandelt werden. Diese Arbeit stellt Ansätze für die Generierung von Erklärungen in einer Vielzahl von Problemstellungen vor: Es ist nicht nur möglich, die Entscheidung für ein einzelnes Bild zu erklären, es ist auch möglich, das Modell als Ganzes zu erklären. Ebenso können Erklärungen davon profitieren, wenn der Zugang zu den Modell-Parametern gegeben ist.

Um die relationale Erklärungs-Generierung holistisch anzugehen, wird sich diese Arbeit auch mit Methoden beschäftigen, wie man so nahe wie möglich am Verhalten des Original-Modells bleibt und wie man diese "Modelltreue" quantifiziert. Zusätzlich wird diese Arbeit nochmals auf die generierten Erklärungen zurückkommen und die Frage stellen, welche Arten von Erklärungen besonders nützlich für Menschen sind.

Acknowledgments

There are a lot of folks I would like to thank at this point. Without them, you would not hold this thesis in your hand / read it on your device right now. First and foremost, a big thanks goes to my amazing supervisor Prof. Dr. Ute Schmid. Whenever there were thinking barriers or doubts, a heated but fruitful debate helped a great bunch. Having some fleshed-out and clean paper after an overnight writing session with colleagues / co-authors is the best feeling ever. I also want to thank Prof. Dr. Diedrich Wolter and Prof. Dr. Daniela Nicklas for the helpful suggestions during the complete course of my thesis endeavor.

Losing my sanity is absolutely impossible with my wonderful family: Thank you, mom and dad, for awakening in me the passion for science and curiosity and pushing me just the right amount. Thank you, Katja, for always being there for me and having just the right amount of craziness for me to not become crazy. And last, but not least: Thank you, Abdul, for having the biggest of hearts and the best knowledge of the local barber shops. Without you, all the stereotypes of bearded basement dwellers would have come true with me.

Contents

List of Figures

List of Tables

Part I

Synopsis

1
Introduction

1.1 Motivation

The field of deep learning has heavily everted the AI landscape (Goodfellow et al., 2016). Large hierarchical neural networks with many parameters are trained with a huge amount of raw data. To get a grasp of the mere size of modern networks: OpenAI's large language model GPT-3 has 175 billion trainable parameters and was fed with datasets comprising nearly a trillion words (Brown et al., 2020). You can see by the success and widespread usage of these models, that they are extremely effective at their assigned job. In the field of technical medicine, novel deep learning methods can be applied successfully to a variety of tasks, involving analysis of medical images to support practitioners in their diagnoses (Greenspan et al., 2016). In the detection of breast cancer for example, models based on deep belief networks can achieve accuracies over 99 percent (Abdel-Zaher and Eldeib, 2016).

Although these models typically achieve high performance metrics, they have a major flaw: They are mainly black-box, meaning their parameter architecture makes it difficult for practitioners to comprehend the reasons behind their decisions. Merely inspecting the weight matrices does not give an overvalue when trying to understand what the model is doing. This fact is often overlooked in favor of the fact that they work so incredibly well. But the black-box nature is one of the major disadvantages that hinders widespread implementation of ML models, especially in critical fields like medical diagnosis (Watson et al., 2019) or in quality assurance in the industrial sector (Müller et al., 2022). It is essential that experts in a field always have the possibility to audit such models. That way, when the - otherwise well performing - machine makes an error, the personnel can easily gain insights in what, where and why something went wrong. Only this way one can gain trust when using such automatic decision systems. This is crucial when bringing AI systems into the real world.

When searching for solutions, one makes a find in the field of eXplainable Artificial Intelligence (XAI) (Gunning et al., 2019; Adadi and Berrada, 2018). Research in this field aims at introducing mechanisms that make automatic decision systems more transparent to humans by keeping their high level of predictive performance. It is an important goal of researchers in XAI to actively involve the human in decision processes, keeping it from becoming a mere bystander.

Looking in the past, before big neural network driven systems, there were already many transparent approaches. Decision Trees (Quinlan, 1986) or Inductive Logic Programming (ILP) (Muggleton and De Raedt, 1994) build symbolic theories that are in a language close to the natural human languages (Siebers and Schmid (2019) show that e.g. ILP programs can be easily converted into natural language). When auditing the decision making process for a particular instance, it is fairly easy to automatically generate the trace of deductive reasoning. This trace itself is again in a symbolic language and therefore easy to understand for humans. So the question arises: Why don't we just use such systems nowadays for the critical application areas? Unfortunately, such systems are often affected by an accuracy-interpretability trade-off. While they are fairly transparent, their performance metrics on real world problems are rather poor. One main reason for that is the rather bad fit to raw data. In the image context, modern Convolutional Neural Networks (CNNs) perform extremely well in grasping the semantics behind images and their parts. This is achieved by their complex layered structure of matrix operations, specifically designed for numerical input. For ILP, one has to first summarize and symbolize the image and its parts before being able to induce hypotheses. This can not always happen in a way that all important aspects in the image are taken into account.

A possible expedient is the following approach: We do not relinquish the obscure but well-performing neural network models and still use them for the decision making. However, after training is completed, we take them as is and additionally generate an interpretable surrogate model, mimicking the behavior of the original model to the best of its ability. These post-hoc explanations are now easier to audit. The generation of such post-hoc explanations via symbolic surrogates will be the subject of this thesis.

The gold standard in the image domain for such surrogates in the context of XAI are visual attribution methods like LIME (Ribeiro et al., 2016) or Grad-CAM (Selvaraju et al., 2017). Given an image instance and an ML model, these approaches can make visible, what image parts are mainly responsible for a decision for the instance w.r.t. the model. While these methods are a first start in showing users where the model might have abnormal behavior, they lack in expressiveness. What is meant by that can be explained when having a look at how explanations should be communicated to its recipients. Explanations in the context of XAI are inherently a conversation between machine and human. A human has an explanation need that the machine needs to satisfy[1]. The principles of cooperative conversations in the form of the Gricean maximes (Grice, 1975) give clues on what a good explanation has to contain:

- **Quantity**: All required information has to be contained in the explanation.
- **Quality**: The explanation has to be truthful (in our case to the black-box model).
- **Relation**: Irrelevant information should be omitted.
- **Manner**: Be clear of what the explanation expresses. Thus, avoid obscurity and ambiguity.

[1]This exchange of information does not have to be a one-way street (see Rabold et al. (2022)).

For this work it is argued, that especially the maxim of manner falls short in visual attribution methods. Simply highlighting important parts in the image can only convey part of the information that is important to understand models. Especially in the aforementioned critical domains, concepts like fractures in bones or in car parts are of relational nature. There is a big difference in diagnosis and treatment depending on where the fracture occurred, in relation to bone parts. A fracture in the middle of the femur might not be pleasant, but possibly means a cast and a few months rest. A fracture in the femoral neck however can render the need for an expensive prosthesis (especially for elderly people) (Marya et al., 2008). Similarly, in industrial quality control, the spatial relation between a blowhole and another part of machinery could decide between keeping or scrapping a component (potentially saving lives by avoiding car crashes) (Müller et al., 2022).

Such visual-relational domains call for more expressive surrogates. This is exactly where ILP comes into play. The first-order logic nature of the induced models can easily convey relational aspects of learned concepts and does not obscure this information (like the propositional logic nature of the visual attribution methods). This thesis focuses on building and evaluating methods to generate expressive, faithful and useful ILP surrogates for black-box models.

A holistic approach to this problem needs to take different aspects into account: Problem domains come in all shapes and sizes. Therefore, different approaches need to be explored to extract useful information that can be used to build the symbolic relational surrogate. Also, the most expressive surrogate is not useful, if it is not faithful to the original model it mimics (see the maxim of quality above). For this reason, this thesis explores ways of enforcing and quantifying fidelity of the ILP surrogates w.r.t. the black-box. Of course, when it comes to explanations, we can not cancel the human out of the equation. We will have a look at how we can meet different explanation needs for different experts and other users of an automatic decision system. According to the goals of XAI, instead of being a mere bystander, we need to actively involve the human in the explanation process. After all, a machine should never be the last decision instance, but rather be a helpful companion for the final decision maker: the human.

1.2 Research Questions

Based on the considerations of the preceding section, the following main research question will guide the remainder of this thesis:

RQ: *How can we generate expressive, faithful and useful relational explanations for black-box machine learning models in the visual domain?*

In particular, to build ILP surrogates that are expressive, faithful and useful for humans, the following sub-questions will be answered. The questions are accompanied by the respective scientific papers, where details on algorithms, the experimental setting, and results can be found.

SQ1: *How can relevant sub-concepts and their relations be extracted from the to be explained problem domain?*

This question is extensively explored in Rabold et al. (2020a), where a visual attribution method is first used to find regions in images important for the classification result. These regions as well as their relations are then symbolized to form the foundation for relational explanations. Additionally, in Rabold et al. (2020b), the same symbolization is done, but instead of purely working within image space, methods of concept embedding analysis are used to find regions in images where semantic concepts learned by the network are located. Finally, in Rabold (2022), instead of relying on pre-trained concept localization, concepts present in images are found by generalizing a small human annotated set to a larger sample size that is used to localize semantic concepts.

SQ2: *How can a high fidelity of the generated explanation towards the behavior of the original black-box model be established and quantified?*

To answer this question, in Rabold et al. (2018) a modified evaluation function for our relational explanation generation framework is developed and evaluated. This new function takes image similarity of desired instances into account to explain classification in the vicinity of a given sample. Rabold et al. (2020a) showcases the usefulness of visual attribution methods as method of filtering image content that is important for a particular classification result. The fidelity of generated relational explanations w.r.t. the original model as generated in Rabold et al. (2020b) and Finzel et al. (2024) are quantitatively evaluated with a modified accuracy metric.

SQ3: *How can we generate explanations useful for humans?*

In Rabold et al. (2022), an empirical study is conducted and evaluated in order to find out which modes of explanation are particularly useful given different explanation needs of users. Additionally, in Rabold (2022), methods are shown how users can time-efficiently adapt the explanation generation to their own specialized domains without having to rely on pre-designed datasets and concept localization models.

1.3 Outline

The remainder of this thesis is structured as follows: Chapter 2 gives background to the used methods and lists important related work. In Chapter 3, the problem domains used in this work are outlined. Chapter 4 describes the general approach for generating relational explanations and answers the research questions stated above by contextualizing the published work. A conclusion and outlook is given in Chapter 5. The synopsis ends with listing the references of the papers that are part of this thesis in Chapter 6 and stating the contribution I had on them.

2
Background and Related Work

This chapter will be describing the literature background and related work of this thesis. Special focus is set on concepts and frameworks that are used as foundation of the approaches developed in the course of this thesis.

2.1 Convolutional Neural Networks for Image Classification

In earlier days of AI, the field mainly focused on problems that were difficult for an average human to solve efficiently. Some examples include planning in a complex setting of states and actions (Hendler et al., 1990) or playing chess against grandmasters (Campbell et al., 2002). These problems however typically have a quite manageable and understandable formalization and search algorithms can find solutions in their problem spaces (although these spaces typically becoming quite large). Bringing AI out in the real world however required solving problems that the average human (expert) solves with ease (like recognizing faces, understanding language, diagnosing diseases via MRI images etc.)[1]. These tasks however are typically under-formalized and can not be grasped by a set of a few symbolic instructions.

For such tasks, architectures based on neural networks rendered considerately effective. Neural networks are able to make sense of complex raw data and use their hierarchical architecture to extract information in a highly non-linear way. This helps grasping obscure information for example in images that would have otherwise been overlooked by humans.

There is no doubt that the vast field of deep learning (a sub-field of Machine Learning) is currently the most promising approach to tackle these kind of problems on a large, applicable scale. A crisp definition of deep learning is hard to obtain; many sources agree however on neural network architectures forming the foundation of deep learning (Goodfellow et al., 2016; Deng and Yu, 2014). Also, in contrast to shallow learning architectures, deep learning models should have a considerable number of layers (Goodfellow et al., 2016). The hierarchical setup of the layers also takes a central role. Let us take the example of a model processing image data. Most humans have to rely on the visual sense to perceive the world, with visual perception being one of the most important

[1] See also the AI definition given by Rich (1983).

senses. Therefore, it is no wonder one of the biggest research fields are deep learning approaches processing images or video.

Typically, the underlying structure of such models is some form of Convolutional Neural Network (CNN) (LeCun et al., 1989; Krizhevsky et al., 2012)[2]. CNNs are capable of making sense of images by being able to detect hierarchies of visual features within an architecture of stacked layers. A trained deep learning model for image processing contains sets of parameters that act as feature detectors, also called filters. On lower layers (closer to the input layer), these filters might respond stronger to the presentation of simple edges or color blobs. When progressing to higher layers, the filters are more differentiated. Compositions of edges forming simple shapes and textures are now the reason of excitement for the filters. In the final layers of the feature extraction part, whole objects are detected and can be used to infer the final output of the model.

In the following, an overview of the operating principles of CNNs as well as a formalization shall be given. This will act as the foundation for the approaches described in this thesis, which are focused on image input. For a more in-depth view in the mechanisms, the reader is referred to Goodfellow et al. (2016).

As already mentioned, a CNN M consist of a stack of layers $l_1, \ldots, l_m \in M$. There are different types of layers, each having a different purpose of processing the data that is produced by the layer directly before it. The name-giving layer of CNNs is the *convolution layer*. Assume a convolution layer l_i. The output of the layer before it, $o(l_{i-1}) = x$ (which can also be the input image), is convoluted by a set $k_1, \ldots, k_n \in K$ of n 2D kernels of trainable weights, also called filters[3]. In an already trained layer, when feed-forwarding an image through the network, this results in n 2D activation maps with high values at positions where there is a visual feature that is important for the task at hand. A single position $[a, b]$ in the 2D activation map $o_j(l_i)$ for one of the filters $k_j \in K$ is calculated with Equation 2.1, where h is the height of the filter, w is its width and c is the number of channels of the filter. Note that typically, in order to being able to calculate the convolution, the filter is simply duplicated to match the number of channels c in the input image or activation map.

$$o_j(l_i)[a, b] = \sum_{r=0}^{h} \sum_{s=0}^{w} \sum_{t=0}^{c} k_j[r, s, t] \cdot x[a + r, b + s, t] \tag{2.1}$$

Another layer is the pooling layer, whose purpose is to aggregate data and discard unimportant information. Typically, a max pooling is used where the values of the activation maps are separated in blocks and only the highest values in a block are being kept. This again results in (smaller) 2D maps.

When the information in an image is sufficiently compressed and summarized, the final activation maps are converted from 2D into a 1D representation of the image by simply scanning the values line by line. This acts as an input for a standard feed-forward neural network where the gathered information is re-weighed to make a final decision on what the image should be labeled. It is

[2]Bakator and Radosav (2018) highlight, that the majority of methods applied to analysis of medical imaging is based on CNNs.

[3]Note, that actually in many implementations, a cross-correlation between input and kernel is calculated. To stick with the terminology most often used in literature, the term convolution is used instead of cross-correlation.

important to mention that all intermediate output in a neural network is typically being put through an activation function. The most common function is the Rectified Linear Unit (or ReLU) function (see Equation 2.2). An activation function's purpose is to introduce non-linearity in the system in order to make it possible to learn complex correlations between input and output. Figure 2.1 indicates the output dimensions of the typical layers of a sample CNN.

$$f(x) = max(0, x) \tag{2.2}$$

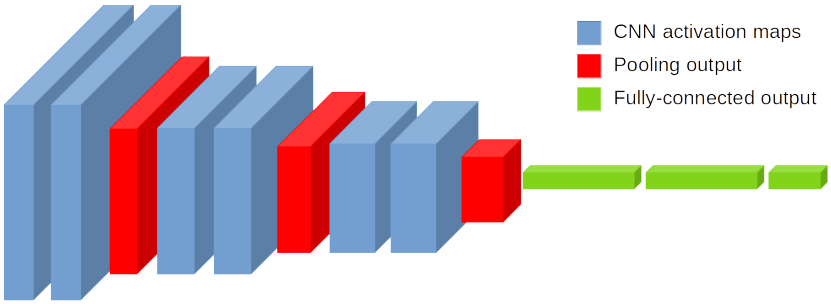


Figure 2.1: Visualization of the respective "output cubes" of the layers of a sample CNN. Convolution layers typically result in multiple stacked activation maps, where the number of activation maps grows when going deeper into the network. Pooling layers reduce the information and make subsequent activation maps smaller. The output of the last pooling layer is flattened into a 1D vector and is used as input for fully-connected layers.

CNNs are typically trained with a large set of labeled images in an end-to-end fashion. A loss function evaluates the match of the output of a CNN with the desired output for a training image and uses back-propagation (Rumelhart et al., 1986) to feed the error back into the network. The layer weights are then updated according to the influence they have on the network output. The influence is calculated by derivatives, e.g. when updating weight w with a calculated network output y, the change of w is dependent on $\frac{\delta y}{\delta w}$. To ensure that back-propagation can alter the weights during training from end to end, all calculations in the network need to be differentiable.

As already mentioned, CNNs along with many other deep learning architectures have great performance metrics when used for raw data. The bigger a network becomes, the better its potential to make sense of highly non-linear concepts[4]. However, the computational effort in training deep networks is immense. In Strubell et al. (2019), the authors concluded that inducing a typical state-of-the-art deep learning model emits nearly five times the carbon dioxide exhaust of a car - over its entire lifetime. Despite the high performance metrics, CNNs can also make errors that are not easily explainable by humans by merely inspecting the trained parameters. A typical error source is bias in the image data selected for training. Ribeiro et al. (2016) make this problem explicit by enforcing a bias in a classifier trained to differentiate between images showing huskies and wolves. Training images depicting wolves were cherry picked such that they always contained snow, whereas the images showing huskies did

[4]See for example He et al. (2016).

not. Consequently, the classifier learned, that the concept "wolf" simply means "snow/light surface is present", without picking up on actual features of a wolf.

The black-box nature of CNNs makes it difficult to just use common sense to see what is wrong with the model. The other pole of this spectrum would be symbolic approaches like Inductive Logic Programming, described in the following section.

2.2 Inductive Logic Programming

Inductive Logic Programming (ILP) (Muggleton and De Raedt, 1994) is a research field combining machine learning with logic programming[5]. It is often labeled as human-like, knowledge-level learning (Dietterich, 1986), since its formulated approaches of inducing hypotheses from observations resemble human reasoning. The field can be seen as stark contrast to statistical machine learning approaches like the neural network driven CNNs (see Section 2.1). Not only does ILP generally need significantly fewer observations (positive and negative examples of a concept to be learned); it also produces its hypotheses in the form of human-interpretable symbolic theories[6].

One of the first applications of ILP in a complex real world domain was conducted by King et al. (1996) to predict the mutagenicity of chemical compounds with significant improvement in comparison to other approaches. This worked so well, because in contrast to other knowledge-level approaches like decision trees (Quinlan, 1986), ILP algorithms use first-order logic (FOL) to describe positive and negative examples. This makes it suitable to induce hypotheses for problems that are of relational nature (such as the topology of molecules). Further, ILP is able to incorporate additional explicit (expert) knowledge in the learning process. This makes it possible to also use custom domain knowledge, users of a machine learning system might already have, actively incorporating field experts in decision processes.

In the following, a short formalization of FOL is given. Let C be a set of constant symbols and V be a set of variable symbols. Terms $\mathcal{T}$ in the sense of FOL are all variables $v \in V$ and all constants $c \in C$. Functions are excluded for now since they are not in the scope of this work. The set $\mathcal{F}$ of well-formed formulae consists of the following:

- $\top \in \mathcal{F}$, $\bot \in \mathcal{F}$
- $p(t_1, \ldots, t_n) \in \mathcal{F}$, where $p \in P$ and $t_1, \ldots, t_n \in \mathcal{T}$. n is called *arity* of p
- $\neg\phi \in \mathcal{F}$, if $\phi \in \mathcal{F}$ (logic negation)
- $\phi \odot \psi \in \mathcal{F}$, if ϕ,$\psi \in \mathcal{F}$, where $\odot \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ (logic junctors with their common interpretations; when the implication arrow $\rightarrow$ is flipped to $\leftarrow$, the implication is read from right to left)
- $Qx.\phi \in \mathcal{F}$, if $\phi \in \mathcal{F}$, where $x \in V$, $Q \in \{\forall, \exists\}$ (logic quantifiers)

[5]For example expressed in the logic programming language Prolog (Clocksin and Mellish, 2003).

[6]Sometimes referred to as white-box learning in contrast to the black-box nature of neural networks.

$p(t_1, \ldots, t_n) \in P$ is called a predicate. A predicate or its negation ($\neg p(\ldots)$) is called a literal. A formula that only contains disjunctively combined literals (e.g. $\phi = p(\ldots) \vee \neg q(\ldots) \vee \neg r(\ldots)$) is called a clause. A clause with at most one positive (non-negated) literal is called a definite Horn clause (in the following just called Horn clause). Horn clauses are the central elements of *theories*, the hypotheses induced by ILP methods. A Horn clause can easily be transformed into a logic implication (we also call these rules) by leveraging the semantic equivalences of implication and de Morgan. As an example, take the Horn clause ϕ from above. It is equivalent to $q(\ldots) \wedge r(\ldots) \rightarrow p(\ldots)$. We call the left side of the arrow the body and the right side the head of the rule.

This work often makes use of the syntax of the logic programming language Prolog (Clocksin and Mellish, 2003). Prolog symbols that start with a lower case letter are considered constants (if they are used as terms) or predicates. Symbols starting with an upper case letter are considered variables. The rule from above would be written like this (with additional constants and variables) in Prolog (note the flipped order of the implication with a stylized "arrow"):

```
p(A,B) :- q(A), r(B,A).
```

Like any machine learning approach, ILP algorithms are inducing a hypothesis H (called *theory*) while considering the conjunction of positive (E^+) and negative (E^-) examples. Examples are given as first order predicates. Additionally, symbolic background knowledge (or simply BK) B in the form of FOL facts and rules, describing the examples or giving domain knowledge is given.

Theory H, effectively being a set of Horn clauses, is constructed such that the following conditions hold (adapted from Muggleton and De Raedt (1994)):

- $B \wedge E^- \not\models \bot$ (prior satisfiability; negative examples are consistent with BK)
- $B \wedge H \wedge E^- \not\models \bot$ (posterior satisfiability; negative examples are also consistent with hypothesis)
- $B \not\models E^+$ (prior necessity; positive examples can not be explained with BK alone)
- $B \wedge H \models E^+$ (posterior sufficiency; BK together with hypothesis explain positive examples)

Over the years, the ideas of ILP were implemented in a variety of frameworks that can all be used for either more or less general purpose applications. An early framework is Aleph (Srinivasan, 2007). Its versatility will aid as a general tool for the approaches developed in this work. In the following, the functioning of Aleph will be discussed. For a detailed description of Aleph, the reader is referred to Srinivasan (2007) and Gromowski et al. (2020).

Induction of a hypothesis is realized with a nested search. In an outer loop, the following steps are taken by the Aleph algorithm (adapted from Srinivasan (2007)):

1. **Example selection**: Pick a positive example $e \in E^+$ as a seed for generalization. If no such example exists, halt.
2. **Best clause search**: *see below*
3. **Cover removal**: All positive examples covered (i.e. modeled by the theory) are removed from E^+. Go to Step 1.

Best clause search: The inner loop takes one example $e \in E^+$ and finds the "best" possible Horn clause by refining the most general clause consisting of only the head without any constants. Refining means adding literals in the limits of a language bias (realized with modes) imposed on predicates. That means, that there can be limitations of whether variables or constants are allowed as arguments and at which positions. The "best" clause is defined by a score. The evaluation function to calculate the score can be adapted. But in the default setting, higher scores are given to clauses that cover many positive examples by avoiding to cover negative examples. The highest ranked clause in an iteration is checked to not cover any negative examples from E^-. If that is the case, the clause is returned to the outer loop for cover removal. It can happen that no clause is found. In that case, the chosen example itself is returned to the outer loop.

There are multiple reasons why ILP is suited for the endeavors of this work. As already mentioned in the motivation for this thesis, the first-order logic representation of the learned rules can explain important inter-relations in visual relational domains prevalent in critical application fields. Visual attribution methods (further showcased in Section 2.3.1) can only give a quick, but incomplete idea of what parts in an image are important for a decision.

Prior work (Muggleton et al., 2018) indicates, that the induced symbolic hypotheses of ILP are inherently useful for humans when it comes to understanding learned hypotheses. Especially when generating explanations for black-boxes, it can be favorable to leave the numerical domain of neural networks and stick to a language that is closer to the natural language humans are used to. Siebers and Schmid (2019) conclude that it is also possible fairly easily to transform the induced first order Horn clauses into natural language sentences using template-based approaches.

In the context of explanations useful for humans, we can use Donald Michies three criteria for machine learning systems (Michie, 1988): While the weak criterion simply states that with more data, the predictive accuracy increases, the strong criterion demands the induced models to be of symbolic form. We can go even further with the ultra-strong criterion. It is met when a machine learner is able to also explain the hypothesis inherently to a human such that the human can profit from the explanation in a way that goes beyond merely studying the provided examples. Muggleton et al. (2018) shows that ILP fulfills all three criteria.

2.3 Explainable Artificial Intelligence

The goal of this thesis is to show methods for building transparent, expressive explanations for otherwise black-box CNNs. This puts this work in the context of eXplainable Artifical Intelligence (XAI) (Gunning et al., 2019). Its goal can be summarized as follows: Building AI systems that keep a high performance, but in parallel are able to communicate the reasoning behind decisions to users in the best way possible (see e.g. Adadi and Berrada (2018)).

To understand why it is important to be able to effectively communicate the inner workings of decision systems to humans, we can have a look at the four dimensions of XAI, according to Adadi and Berrada (2018):

1. **Explain to justify**
   XAI systems enable users to audit automatic decisions. Especially when a black-box system acts erroneous/biased, XAI approaches help in identifying the reason for this abnormal behavior. Ultimately, this helps in establishing trust in such systems. Additionally, justification of automatic system behavior is necessary to be in compliance with legal requirements, such as the General Data Protection Regulation (GDPR) (Directorate-General for Communication, 2023).

2. **Explain to control**
   Having a transparent system (or a black-box system with a transparent interface) helps in better understanding an automatic system from early stages on. This helps in counteracting errors early in development, before systems are deployed in critical application fields.

3. **Explain to improve**
   XAI modules also foster improvement of ML systems. By understanding the function between input and output, users get to know possible points where a system can be improved.

4. **Explain to discover**
   Modern ML systems are exceptionally good at understanding complex domains. It is of great value to be able to make the learned knowledge explicit by means of XAI methods. This way, practitioners can learn even more about a particular domain and use this knowledge in future endeavors.

These dimensions always keep in mind the recipient of machine generated explanations: The human. Explanations can simply help humans in debugging an erroneous system and improving it. Or they can even aid better understanding of a complex domain and thus, help humans find new ways of developing life saving drugs or tackling the challenges that come with climate change. This however calls for a tight integration of machine and human. As already laid out in the motivation for this work, explanations are essentially a communication between two agents. Researchers from the social sciences draw many comparisons between human-to-human and machine-to-human interaction (Miller, 2019). A main point is the *contextuality* of explanations. They need to be adapted to the needs of the recipient as well as the problem domain.

XAI approaches are manifold. A comprehensible taxonomy for approaches can be found in Schwalbe and Finzel (2023). The methods created to answer the research questions of this work can be categorized accordingly (only selected means of classification are described):

- **Problem definition**
  This thesis focuses on explaining black-box models (mainly CNNs) classifying images. Models will not be re-trained interactively, but a post-hoc explanation in form of an interpretable surrogate model will be generated.
- **Explanator output type / Presentation**
  What is the representation of the explanation? While explanation means are manifold (contrastive, prototypical, graphical), this work focuses on generating first-order logic rules with the possibility of combining them with the visual input.
- **Portability**
  Portability answers the question, whether it is important to know what type of black-box model will be explained. This thesis presents systems where only the relation between input and output can be observed (model-agnostic) as well as systems where a full inspection of the layer parameters of CNNs is possible (model-specific).
- **Locality**
  A distinction is made between global and local explanations. A global explanation is concerned with how the model in its entirety behaves. Local explanations focus on the question, why a particular decision was made for a single instance. Both aspects will be examined in the remainder of this work.
- **Metrics to evaluate the explanation**
  This work aims at optimizing objective as well as subjective metrics of explanation quality. Approaches to maximize fidelity of explanation models w.r.t. the original model are discussed. Additionally, human preferences of different explanation means as well as methods to efficiently integrate humans in the explanation generation are introduced.

2.3.1 Visual Attribution Methods

Local explanation methods are often realized by attribution methods. The goal of such methods is to quantify the contribution of input features to the final decision of a model. For visual domains, this results in a heatmap that can be applied to the input image in order to see which parts of the image positively or negatively impacted the classification. In the following, two important visual attribution methods will be introduced:

The LIME approach (Ribeiro et al., 2016) is able to highlight elements of an input, that contribute most to the decision of an arbitrary machine learning model. It can be applied to images, text and tabular data. In this work, we will focus on image data. The goal of LIME is to approximate the non-linear decision boundary of the original model in the vicinity of a sample by a linear model. LIME does not directly act on pixels, but on groups of semantically

similar pixels called superpixels, effectively simplifying the representation of an image[7].

A sample set of alterations of the original image is generated by randomly switching off superpixels (by blacking them out or replacing them with the average color of the superpixel). All samples from the set are now re-classified by the original model. A modification of the LASSO method (Tibshirani, 1996) from statistics is able to infer the influence of each superpixel on the original classification and finds a linear model where each superpixel receives a weight. This weight can either be positive or negative, depending on whether the superpixel is important to the decision or counteracts it. The absolute value indicates how strong the influence is.

LIME also considers the similarity of samples to the original image. The more similar a sample is, the stronger the influence on the calculation of the linear model it has. Figure 2.2 visualizes the linear approximation by sampling in the vicinity of a pivotal image. The goal of LIME in general is to minimize the loss stated in Equation 2.3 (Ribeiro et al., 2016). $\mathcal{L}$ is a measure of unfaithfulness of linear explanation model g w.r.t. original model f in the vicinity of classified sample x defined by similarity measure π. Ω is a complexity measure of linear model g and measures the number of superpixels used in the explanation. LIME therefore aims at finding an explanation g that is closest to the behavior of f by minimizing its complexity (and therefore the effort a human has to put into its interpretation).

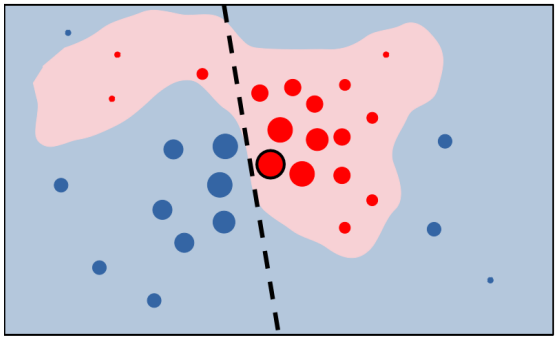


Figure 2.2: Linear approximation in the vicinity of a sample to be explained (with thick outline). The non-linear decision surface of the original model is indicated by the separation of red and blue space. The dashed line is the linear surrogate. Influence on the surrogate generation is dependent on the similarity to the to be explained sample. More similar/nearer samples are drawn bigger (Figure adapted from Ribeiro et al. (2016)).

$$\mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2.3}$$

It is often advantageous to have access to the parameters of a neural network. Model-specific approaches can explain for example, which filters on a layer have the biggest influence on the final classification, which in turn can pinpoint very precisely where in the model erroneous behavior emerges. Grad-CAM (Selvaraju

[7]See e.g. Schallner et al. (2020) for a comparison of superpixel methods in a real world application.

et al., 2017) (short for Gradient-weighted Class Activation Mapping) analyzes the activations in given layers of a network and can project the regions in the original image, that were most influential to the network decision. In contrast to LIME, Grad-CAM is not bound to superpixels and the resulting heatmaps can be arbitrarily distributed over the image. Also, particular filters in network layers can be associated with semantic visual concepts and are therefore often better suited to find building blocks for human understandable explanations. More on this in Section 2.3.2.

While its predecessor CAM (Zhou et al., 2016) is dependent on a specific network architecture (with just one fully-connected layer after the last convolution layer), Grad-CAM can be applied to an arbitrary differentiable convolution network. Let y^c be the output of a network for class c (before any class normalization like softmax). To find the influence of particular positions in the k feature maps $A^l_{1,\dots,k}$ in layer l on the output for class c, calculate the gradient maps $\frac{\delta y^c}{\delta A^l_i}$ where $i \in [1, \dots, k]$. This leaves you with k 2D maps of gradients. Next, receive a single value α^c_i by simply finding the average value for each gradient map. Finally, aggregate the values for each feature map to form one single 2D map by applying Equation 2.4. The ReLU function (see Equation 2.2) is applied to filter out negative (and therefore unimportant) regions.

$$\text{Grad-CAM}^c = ReLU(\sum_i^k \alpha^c_i A^l_i) \tag{2.4}$$

The result can then be up-sampled to match the dimensions of the input layer. That way, a heatmap is formed to indicate regions deemed to be important for class c.

As already mentioned in the motivation for this work, visual attribution alone only highlights what constituents are important for a deep learning model's decision. In visual-relational domains, a classification is dependent not only on what objects are present, but in which constellations these objects are. Prior work (Rabold et al., 2020a,b) indicates that CNNs are capable of distinguishing between classes, where the objects in the images are the same, but in different spatial relations to each other. Simply highlighting two important objects is just not expressive enough to explain to the user, that the relation between them was important for a class decision. It is important to mention that there are already existing XAI that output explanations based on symbolic rules (Ribeiro et al., 2018; Guidotti et al., 2018a). However, these explanations are not relational, but simple if-then rules that merely check attribute values. This does not make them more expressive than a simple propositional decision tree. This thesis shows ways of how ILP methods can be used to generate expressive relational surrogate models, highlighting the importance of not only image constituents, but also their relational constellation.

2.3.2 Concept Embedding Analysis

To fully understand the processes inside a neural network, it is helpful to be able to gain access to its parameters. We might not simply be able to look at the parameter values and understand what concepts are encoded. However, methods of *concept embedding analysis* help in shedding light upon the role of filters in particular layers of CNNs and which concepts they detect.

To understand the rationale behind concept embeddings, it is helpful to have a look into research of automatic text processing. When trying to make text manageable for neural networks, the words in a text document are often converted to word embeddings (see e.g. Mikolov et al. (2013)). An analysis of the text corpus finds vectors of fixed size for each word in the vocabulary. A vector assigned to a word is its word embedding. The vectors are not random, but are generated in such a way, that words that share a similar meaning, have vectors that are close to each other w.r.t. a distance metric like L2 or cosine distance.

In model-specific explanation situations, we want to know, which information the network has learned. In particular, for CNNs, the goal is to find out which visual concepts the network is able to detect. Just as in text processing, we can find vectors that correspond to concepts in a fixed pool of concepts (the analogy to the vocabulary). But unlike in the text domain, these vectors are not simply dependent on the statistics of the data[8]. They need to be generated as seen under the lens of a neural network. One of the first methods comes from Fong and Vedaldi (2018). Their Net2Vec approach examines the so called *feature vectors*, i.e. $o_{[1:n]}(l)[a, b]$, the vectors spanning all n feature maps of a layer l at a particular position $[a, b]$ when feed-forwarding an image through the network (see 2.1 for the formalization of CNNs and Figure 2.3 for a visualization). Earlier work (Bau et al., 2017) focused on relating semantic visual concepts with single units in a network layer. The authors of Net2Vec concluded in their experiments however, that visual concept detectors are distributed over multiple filters. Their work suggests methods to find concept embedding vectors w of size n for concepts, by training the weights $w \in \mathbb{R}^n$ of a fully-connected layer on a concept segmentation task. The input of this task are feature vectors and the ground-truth comes from a dataset of per-pixel labeled images containing a variety of semantic concepts[9]. Each concept embedding vector w is trained by first assuming a linear combination between it and the feature vectors of a layer. This linear combination acts as a segmentation mask when upsampling its results to the size of the original image. The weight itself is then indirectly trained with a per-pixel binary cross entropy loss between the result of the mask and the ground-truth per-pixel annotation provided by the aforementioned dataset. The detailed design decisions of this approach can be found in the original work (Fong and Vedaldi, 2018).

As already pointed out, training classifiers for concept detection requires additional annotated data. Typically in basic research, general purpose datasets with a wide variety of visual concepts like the Broden dataset (Bau et al., 2017) are used. However, when experts from specific domains need to examine decision models, these sets might be too inaccurate or might simply not contain the needed concepts. This thesis will give answers to this problem and show methods of letting users define themselves which concepts they need and letting them pick examples directly in the data.

[8]There exist many approaches that purely work on the data, like SIFT (Lowe, 2004), HOG (Dalal and Triggs, 2005), etc. This is however not the focus of this work.

[9]Net2Vec uses the Broden dataset (described in Bau et al. (2017)) for this.

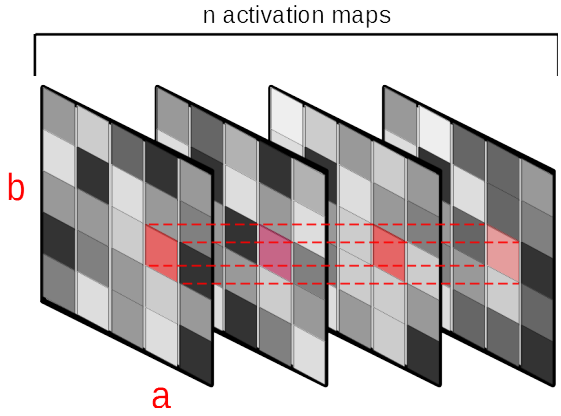


Figure 2.3: An arbitrary feature vector located at position $[a, b]$ in the activation maps of a CNN layer.

2.4 Neural-symbolic Integration

This thesis aims at combining a sub-symbolic neural network model with a symbolic first-order surrogate explanation learned by ILP. The goal here is of course to get the "best of both worlds": A highly effective base model that can learn from raw data, robust to noise, and a symbolic model that gives the ability to reason over the behavior of the base model in a human understandable way. This is exactly the aim of the field of neural symbolic integration (NSI) (d'Avila Garcez et al., 2019) – or neural symbolic computing – and accordingly, this work is also located under this generic term.

There are multiple perspectives on NSI. Generally, as the name suggests, the architectures include a more or less strong combination of a neural and a symbolic part. Classic architectures like KBANN (Towell and Shavlik, 1994) and CILP (d'Avila Garcez et al., 2002) directly integrate logic propositions into a differentiable neural-network-like structure, modeling logic inference rules. By only employing propositional logic, such systems are incapable of explaining relational domains as already discussed earlier. Tensorization approaches like logic tensor networks (Serafini and d'Avila Garcez, 2016) or δILP (Evans and Grefenstette, 2018) incorporate symbolic knowledge in a differentiable architecture by finding vectorial embeddings for constants and realizing predicates and functions as tensors. While it is possible to represent relations with such systems, they can not be used to post-hoc explain already trained conventional architectures like CNNs.

An alternative way of fusing neural networks with logic rules comes with DeepProbLog (Manhaeve et al., 2018). The name is an amalgamation of three principles: Deep learning, probability theory and logic programming. Let us first dive into the combination of the latter two. ProbLog (Raedt et al., 2007) is an extension of the logic programming language Prolog (Clocksin and Mellish, 2003). ProbLog programs, just as Prolog programs, consist of clauses (facts and rules). Additionally for each clause, a probability of it being true is given. By considering the probability distributions over possible programs, given the probabilities of the clauses, a typical task in ProbLog consists of finding the

probability, that an issued query succeeds. This principle is well suited for incorporating neural networks into symbolic reasoning. DeepProbLog (Manhaeve et al., 2018) allows for facts where the probabilities stem from the output of a neural network. An example would be a predicate `digit(X, X2)`, that takes an actual image of a digit (like for example from the MNIST dataset (Deng, 2012)) and a single one-digit number. The probability distribution of this predicate depends on the output of a neural network that is trained on classifying digits from an image. It is important, that the last layer of this network is normalized by a softmax activation function (see Equation 2.5, where z is the non-normalized output of the neural network and C is the number of classes), that outputs values between 0 and 1 in order to fit the probability context. For a well-trained network, a plausible probability of `digit(`  `, 9)`, would be close to 1, whereas for `digit(`  `, 9)` it would be closer to 0.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.5}$$

The DeepProbLog approach comes closest to the neural-symbolic methods pursued in this thesis. However, there are important distinctions:

- While DeepProbLog trains probabilities for its already fixed rule set along with the black-box network, this thesis shows a post-hoc explanation model where a logic surrogate is generated after black-box training is completed. It is argued that this approach is better suited as a mean of explanation in real world applications, since a set of probabilistic facts and rules might not always be readily available.
- In DeepProbLog, dedicated black-box models are used as source for fact probabilities. When there are domains with a large variety of concepts (not only digits, but object parts, textures, colors etc.), the amount of additional computational weight can become quite big[10]. In this thesis, methods are introduced that use information contained in one already trained black-box network to explain.

[10] Apart from the fact, that the labeled training data for such additional networks might also not be available.

3
Domains

There are several domains used to showcase the effectiveness of the explanation generation approach. They range from constructed toy data to real live images. A focus is set on domains where it is imperative to not only have an explanation via visual attribution but also a verbal one. So most of the underlying classes can only be explained by stating attribute values and relations between the building blocks in the image. This chapter showcases each domain and explains what the properties are that make it suited for this thesis.

3.1 Ancient Graves

The Ancient Graves domain (Rabold et al., 2018) consists of generated sparse black-and-white images of stylized grave stone formations. This domain is introduced as a proof-of-concept to be able to quickly generate images where the symbolic representation as well as the classification is given by construction. The distinction between graves coming from either the iron age or the viking age is arbitrary and dependent on the form of the grave as well as the relation between stones.

The four attributes we use in our implementation are

- form: `narrow` or `round` (`narrow` if axis ratio < 0.5)
- number of stones: `many` or `few` (`many` if #stones > 10)
- corner stones: `thick` or `normal` (`thick` if circumference $>$ 2x average size)
- orientation: `north` or `west` (`north` if angle between -45 and +45 from vertical)

The domain is extended to also include relations by introducing a horizontal row of stones with the five possible sizes `smallest`, `small`, `medium`, `large` and `largest`. The sizes of these inner stones is random. Figure 3.1 shows examples from this domain.

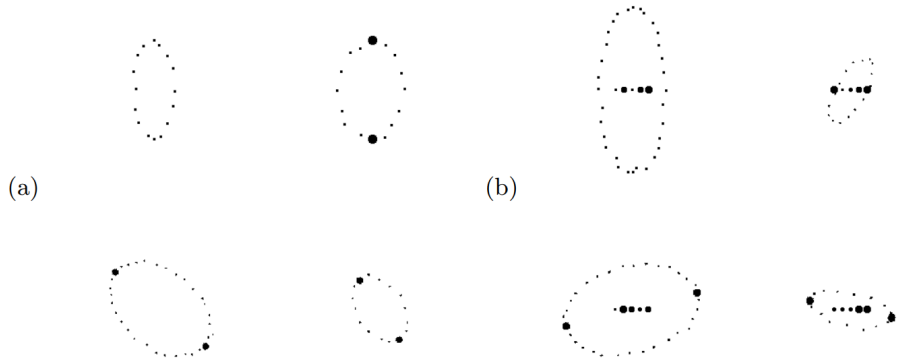


Figure 3.1: Examples for the Ancient Graves domain in the basic propositional setting (a) and in the relational setting (b); top row are positive, bottom row are negative examples (from Rabold et al. (2018)).

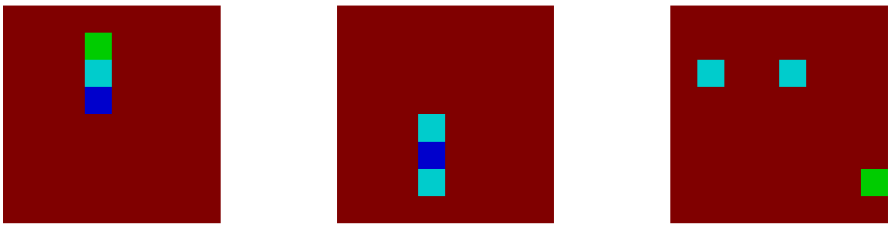


Figure 3.2: Examples from the Blocksworld domain (from Rabold et al. (2020a)).

3.2 Blocksworld

This domain consists of 32x32 color images where monochromatic tiles (= blocks) are arranged in a regular 8x8 grid. The background is colored red and the blocks' colors can be either cyan, green or blue. This simple domain has two perks that come in handy for the endeavors of this work: First, it is easy to automatically extract symbolic color information for the blocks. In our experiments, we simply calculated the most similar named color w.r.t. the block color in RGB color space and used this name as symbolic representation. Second, the regular grid is particularly useful when working with CNNs, since the size of filters in the network as well as the hyper-parameters for the pooling layers can be set to match the block size. That way, neurons in the downstream of the network are specific for a block and the inter-play between blocks can be examined better. Figure 3.2 shows examples of the Blocksworld domain.

Figure 3.3: Exemplary Picasso faces in either a normative (left) or diverging (right) arrangement of facial features (from Rabold et al. (2020b)).

3.3 Picasso

The Picasso domain features 224x224 color images with human faces. The eyes, the nose and the mouth are either in their normative position (positive example) or are mixed up (negative example). Note, that there will always be two eyes, one nose and one mouth. The features are also not just somewhere in the face, but are placed at the original positions (an eye could be at the position of where usually the nose would be).

To create this domain, the FASSEG dataset (Khan et al., 2015) is used. Along with each frontal photo of a face, this dataset comes with masks indicating the occurrence of facial features. The position of features to later place arbitrary features is simply calculated as the center of the bounding box for each mask. To create images where the facial feature arrangement differs from the norm, the features were repainted manually with patches of skin. The original features were cut out before, according to the available masks and kept as a pool to draw from. To create a new shuffled image, features were taken from this pool and simply placed at the positions calculated prior. We made sure to not just swap the left eye for the right eye to create images that truly diverge from the norm. Figure 3.3 shows examples of Picasso faces.

The Picasso domain was created to fulfill three main purposes: First, Picasso serves as domain of images closer to real world images. Compared with a simple blocksworld, objects in the image have more variety than just being monochromatic and we are not dealing with a simple regular grid anymore. Second, Picasso is a domain where the classification depends on spatial relations. This leads to meaningful relational explanations where humans can easily check, if the typical "face arrangement" holds. And finally, Picasso comes with precise pixel-wise annotations of semantic concepts. These can be located automatically on unseen instances via concept embedding analysis (see Section 2.3.2) and used for automatic symbolization of images.

3.4 Family

The Family domain is a simple abstract relational setting of female and male persons connected by family relationships. Figure 3.4 depicts the family tree. In the following, the attributes and relations between the persons is listed as FOL literals:

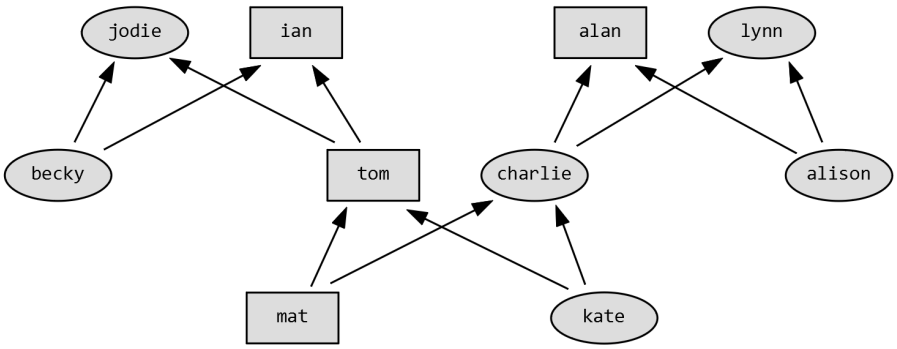


Figure 3.4: The kinship tree of the Family domain. Ellipses indicate `female` persons, boxes indicate `male` persons. The arrows indicate a `parent` relation (adapted from Rabold et al. (2022)).

```
female(jodie)     parent(jodie, becky)
female(lynn)      parent(jodie, tom)
female(becky)     parent(ian, becky)
female(charlie)   parent(ian, tom)
female(alison)    parent(alan, charlie)
female(kate)      parent(alan, alison)
male(ian)         parent(lynn, charlie)
male(alan)        parent(lynn, alison)
male(tom)         parent(tom, mat)
male(mat)         parent(tom, kate)
                  parent(charlie, mat)
                  parent(charlie, kate)
```

The family domain poses a simple abstract relational domain that was mainly used for the research on near miss explanations (Rabold et al., 2022), but also for evaluating, which type of explanation is needed for which domain.

3.5 Arches

In his PhD thesis, Patrick Winston introduced the relational Arches domain (Winston, 1970). He argued, that learning relational concepts from examples benefits from the selection of "helpful" negative examples (in contrast to just random negative examples). A characteristic of these helpful examples is its high structural similarity to positive examples. Just a few (relational) characteristics of these examples prevent it from being part of the positive concept. Winston (1970) called these negative examples *near misses*. Figure 3.5 shows our depiction of positive and negative examples for the Arches domain; the following list gives relations that hold for these structures as FOL literals:

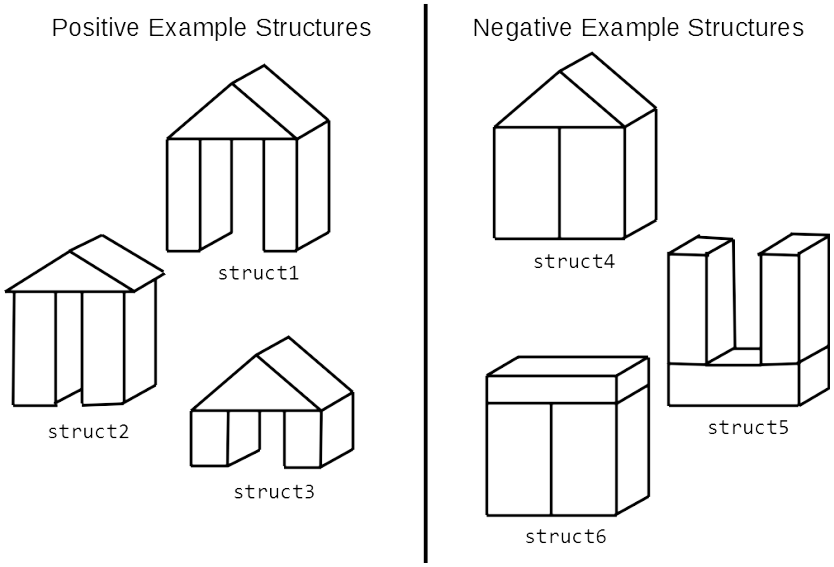


Figure 3.5: Positive and negative example depictions of the concept *arch* (from Rabold et al. (2022)). Note that the negative examples are near misses.

`contains(struct1, a1)`	`contains(struct4, a1)`
`contains(struct1, b)`	`contains(struct4, b)`
`contains(struct1, c)`	`contains(struct4, c)`
`supports(b, a1, struct1)`	`supports(b, a1, struct4)`
`supports(c, a1, struct1)`	`supports(c, a1, struct4)`
`not_meets(b, c, struct1)`	`meets(b, c, struct4)`
`contains(struct2, a1)`	`contains(struct5, a2)`
`contains(struct2, b)`	`contains(struct5, b)`
`contains(struct2, c)`	`contains(struct5, c)`
`supports(b, a1, struct2)`	`supports(a2, b, struct5)`
`supports(c, a1, struct2)`	`supports(a2, c, struct5)`
`not_meets(b, c, struct2)`	`not_meets(b, c, struct5)`
`contains(struct3, a1)`	`contains(struct6, a2)`
`contains(struct3, b)`	`contains(struct6, b)`
`contains(struct3, c)`	`contains(struct6, c)`
`supports(b, a1, struct3)`	`supports(b, a2, struct6)`
`supports(c, a1, struct3)`	`supports(c, a2, struct6)`
`(not_meets(b, c, struct3)`	`meets(b, c, struct6)`
`is_a(a1, wedge)`	`is_a(a2, brick)`
`is_a(b, brick)`	`is_a(c, brick)`

Additionally the "inverse" of the `supports` relation is given by

$$\texttt{supports(Y, X, A)} \rightarrow \texttt{supported\_by(X, Y, A)}\ .$$

The concept of an arch is defined by the following rule (Note the flipped implication for better readability):

$$\texttt{arch(A)} \leftarrow$$
$$\texttt{contains(A, X)} \wedge \texttt{contains(A, Y)} \wedge \texttt{contains(A, Z)} \wedge$$
$$\texttt{is\_a(X, T)} \wedge \texttt{is\_a(Y, brick)} \wedge \texttt{is\_a(Z, brick)} \wedge$$
$$\texttt{supports(Y, X, A)} \wedge \texttt{supports(Z, X, A)} \wedge \texttt{not\_meets(Y, Z, A)}$$

So an arch is a structure that contains two pillar bricks that do not meet and that support a block of arbitrary shape.

This domain is particularly useful for research on near miss explanations in a visual relational domain (Rabold et al., 2022). Just as the Family domain, it is used to research the explanation need for humans in different domains.

3.6 Dogs vs. Cats

A very simple dataset from Microsoft® Research with RGB images showing either a dog or a cat "in the wild". This work actually uses a subset provided by Kaggle. The original dataset comprises over 3 million photos. The subset contains 12,470 dog and 12,491 cat images. This simple dataset is well suited to examine concept extraction in situations, where the classes are very similar (both classes are quadruped, furry animals) and the network has to generate rather specialized concept detectors. Example images and the dataset itself can be found under this URL: `https://www.kaggle.com/datasets/karakaggle/kaggle-cat-vs-dog-dataset` (Uploaded by Karansinh Padhiar; updated 1st of May, 2020; accessed 29th of September, 2023).

4
Generating Expressive, Faithful and Useful Explanations

Explanations are manifold. As already discussed in Section 2.3, explanations have to adapt to the availability of weights and architecture inside the ML model in question (model-specific or -agnostic approaches). Also, it makes a difference, whether single decisions are explained or the overall model behavior is audited (local or global approaches). A holistic view has to consider all of these modalities. This thesis aims at discussing ideas of how expressive explanations for black-box machine learning models can be generated, based on all these different situations. The chapter will first lead through the foundational pipeline of explanation generation. This is followed by a discussion about approaches to extract relevant information from the problem domain that then can be symbolized to build the foundation of relational explanations. I will state the resulting explanations from the experiments conducted in the papers to show that they are reasonable within the problem domain. As a follow-up, the question is raised and answered, how fidelity of a generated surrogate explanation model can be enforced and quantified. Finally, an argument is made about how the showcased explanation generation approaches are useful for satisfying the manifold explanation needs of human users.

4.1 General Explanation Generation Approach

The approaches presented in this work generally follow a simple pipeline as visualized in Figure 4.1.

The behavior of a fully trained black-box machine learning model (1) has to be explained post-hoc. This can either be done for the decision of a single image (local explanation, 2) or for the general model behavior (global explanation). In each setting, a sample of images (3) has to be created, either by generating "useful" alternative versions of a to be explained image (local setting) or by selecting representative image instances (global setting) from a domain. From this sample of positive and negative examples (distinguished by the model decision (4) of the original model for them), visual concepts can be localized and symbolized (5). The decision which concepts are extracted can either come from the model decision (by attribution methods; see Section 2.3.1), from the network

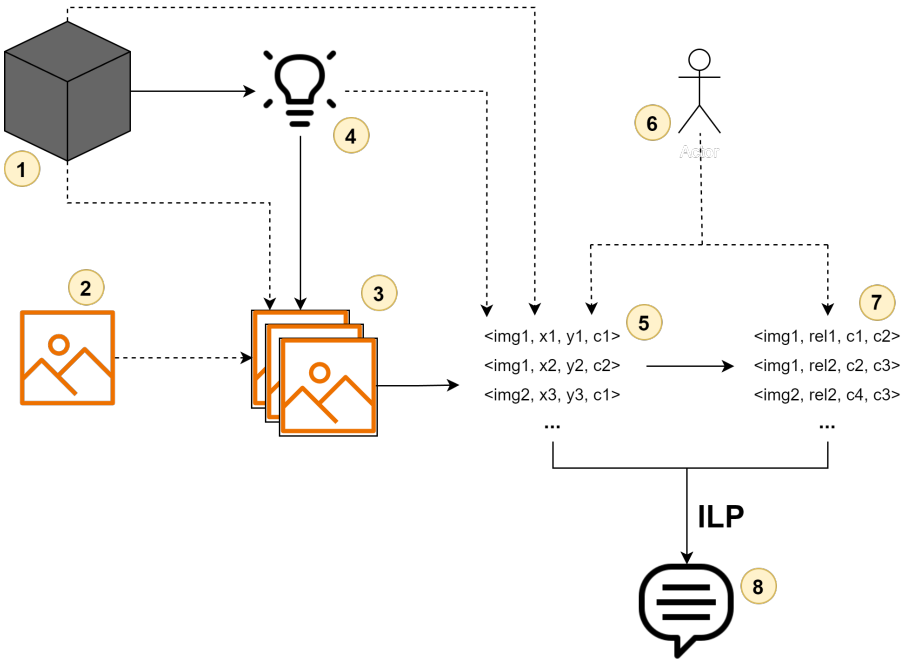


Figure 4.1: The general pipeline of the approach discussed in this work. Constituents: (1) A fully trained black-box convolutional neural network. (2) A single image instance. (3) A sample of positive and negative images. (4) A model decision for one or multiple images. (5) Locations of semantic concepts in images. (6) The human (expert). (7) Relations between concepts. (8) The relational explanation in the form of FOL rules.

filters (by concept embedding analysis; see Section 2.3.2) or from a human (6). With the locations, relations between concepts (7; possible relations are given by a human) can be extracted and symbolized to form background knowledge for an ILP algorithm. The ILP algorithm (mainly Aleph in this thesis) then induces a verbal explanation (8) in the form of FOL rules.

Preferably, the explanation should speak the language of the domain and thus, the users of the system. That means, local explanations should make use of the image at hand to show users what exact parts and constellations in the image led to a decision. This way, flaws in models can be easily detected (as in the "wolf" example discussed in Section 2.1). Conventional visual attribution methods like LIME or Grad-CAM simply highlight (super-)pixels to show important image parts. With visual relational domains, where the classification depends on inter-relations of constituents, we need to transform these important parts and their constellations in a human interpretable form that describes the essential information that led to a decision.

In Rabold et al. (2018), we demonstrate a first idea of how expressive explanations using ILP can be generated. As an example domain, we created a fictional domain of ancient graves, over which we defined a concept according to the Medin and Schaffer concept acquisition task from cognitive psychology (Medin and Schaffer, 1978). The domain is explained in full detail in Section 3.1. A

concept in the context of Medin and Schaffer (1978) is defined by the configuration of binary attributes (so either the attribute holds or it does not). Three propositional rules with a conjunction of two attributes each can be used to define the concept. We arbitrarily declared an iron age grave `X` to be present when one or more of the following Prolog rules hold:

```
iron(X) :- narrow(X), north(X).
iron(X) :- narrow(X), thick(X).
iron(X) :- thick(X), north(X).
```

You can already see that the domain is more complex than to be explainable merely by the highlighting of image parts. Especially for the `narrow/round` property of the outer ring, it would be of no use to just color the complete ring as being important for classification, since it would mean highlighting the complete image. A symbolic statement of what exactly is important for e.g. the ring, is more helpful to grasp the concept of `iron`.

In a second phase, we introduced stones being contained inside of the outer circle. These stones can have one of the following five sizes (ascending): `smallest`, `small`, `medium`, `large`, `largest`. Domain rules made sure that the semantics of a predicate `larger(`a`, `b`)` is met w.r.t. this order. Now, for a grave to be labeled as `iron`, one or more of the following rules have to hold:

```
iron(X) :- outer(X, A), next(A, B), next(B, C), larger(A, B),
                        larger(B, C).

iron(X) :- outer(X, A), next(A, B), next(B, C), next(C, D),
                larger(B, C), larger(C, D).

iron(X) :- outer(X, A), next(A, B), next(B, C), next(C, D),
           next(D, E), larger(C, D), larger(D, E).
```

These rules represent a consecutive sequence of three of the five stones inside the circle that grow in size. The `outer` predicate indicates a pivot stone inside the outer circle. The `next` predicate defines which stones lie next to each other. The rules now feature relational concepts that need to be explained in a symbolic way, since the mere highlighting of the two constituents does not convey the type of relation that holds between them.

As already mentioned above, for this first proof-of-concept, we used a generator to produce our grave images. For each image, the four attributes described above are randomly set to either one of the two values. Also the size of the five stones for the second phase are set randomly. That way, all construction information (attributes and relations) is becoming part of the background knowledge. The images are then filtered according to the constructed rules described above to generate a dataset of iron and viking graves. This filtering process practically takes the place of classification.

As a first experiment, the non-relational graves domain was examined. An example instance x_1 was labeled as `iron` by construction. Additionally, 9 other iron graves and 10 viking graves were sampled. Since the symbolic FOL representation of the graves is already given by the generator, we could easily use Aleph to induce a general rule for why x_1 was labeled as `iron`. Note, that in analogy to LIME, we also took the image similarity of all samples w.r.t. x_1 into

account. This way, we can explain the behavior of the original model (in this case our constructed oracle) in the vicinity of x_1. More on this in Section 4.3.

When instantiating x_1 with a grave with the symbolic representation `narrow`(x_1), `thick`(x_1), `north`(x_1) and `few`(x_1), the following rule was induced:

```
iron(X) :- north(X).
```

We can see that this is not one of the construction rules of an iron grave, but it is close. Another instance x_2 with the representation `narrow`(x_2), `many`(x_2), `thick`(x_2) and `north`(x_2) yielded the following set of rules:

```
iron(X) :- north(X).
iron(X) :- narrow(X), thick(X).
```

This time, all relevant attributes for an iron grave are present in the rules.

We can also learn the means of exclusion from the concept of `iron`. An example instance x_3 with representation `narrow`(x_3), `many`(x_3), (where the corner stones were not `thick` and the grave was not oriented `north`) led to the rules

```
iron(X) :- north(X).
iron(X) :- narrow(X), thick(X).
```

showing the features that have to be absent for it to be labeled as `iron`.

For testing the explanation capability in our relational domain, we generated an iron grave x_4 with the following symbolic representation:

`many`(x_4), `thick`(x_4), `north`(x_4), `outer`(x_4, a), `next`(a, b), `next`(b, c), `next`(c, d), `next`(d, e), `medium`(a), `smallest`(b), `smallest`(c), `small`(d), `largest`(e)

As a result of picking this instance with 9 other random positive and 10 negative instances, Aleph induced the following two rules:

```
iron(X) :- outer(X, A), next(A, B), next(B, C), next(C, D),
           larger(C, D), next(D, E), larger(C, E).
iron(X) :- outer(X, A), next(A, B), next(B, C), larger(B, C),
           next(C, D), larger(C, D).
```

The first rule shows a possible over-specification on the task at hand that might come from unlucky sampling. The `larger` relation is used, but instead of showing the importance of a sequence of stones increasing in size from west to east, the rule simply states that there need to be two stones of larger (but arbitrarily ordered) size on the east of a pivot stone. The second rule however exactly matches one of the construction rules.

The result of this experiment in the relational domain not only shows the interpretability of ILP induced explanations. You can also see how it generalizes away irrelevant information. In the relational domain, the four irrelevant attributes (roundness of the grave, thickness of the stones, orientation and stone amount) do not play a role in classifying the graves. The resulting rules only state the important relations that have to hold between stones.

4.2 Extracting Information for Relational Explanations

In order to adapt to different states of model portability, different methods of information extraction have to be applied in order to receive the building blocks for symbolic relational explanations. In the following, model-agnostic as well as model-specific approaches for extracting relevant sub-concepts and their relations are described and thus, research question **SQ1** is answered.

4.2.1 Model-agnostic Relational Explanation Generation

The ancient graves domain is a toy dataset. Images can be easily generated by randomizing parameters and automatically rendering sparse black-and-white images. For a proof-of-concept, no real machine learning model was trained and the model output was simulated by observing in which category the parameters fell[1]. In a follow-up study (Rabold et al., 2020a), we were mainly interested in pushing towards approaches that can explain decisions of real trained models. Additionally, images typically do not come with a crisp annotation of which objects are present in it and where they are located. So in this case, we need to think about ways of symbolizing important information present in image instances automatically.

Before we dive into this work, there has to be an important distinction made between the field of Computer Vision and the extraction of symbolic information for the purpose of eXplainable AI: When simply summarizing an image with approaches like SIFT (Lowe, 2004) or HOG (Dalal and Triggs, 2005), we do not need to keep an already trained ML model in mind, since these methods are typically a way of generating a numerical input for training sub-symbolic ML models in the first place. In our case however, the selected image parts need to be *important* for an already made decision. This is why Rabold et al. (2020a) uses the visual attribution method LIME (see Section 2.3.1) to find these important image parts.

In fact, the method we use to explain a real black-box model's decision is an adapted form of the LIME approach: The high-level steps of this model-agnostic, local explanation method is as follows (also depicted in Figure 4.2):

- Find regions in the image in question that are highly important for the decision made by a black-box model. This is done with standard LIME for images.
- Automatically extract image attributes from and relations between the found regions and symbolize them.
- Construct altered samples of the original image by finding the inverse of all relations and change them in the background knowledge as well as in the image space.
- Build positive and negative symbolic training examples for Aleph by letting the original model classify the altered images (in addition to the already classified original image).

[1] Or in other words: The ground truth is identical with the model output.

- Let Aleph induce a set of rules for explaining the decision for the image.

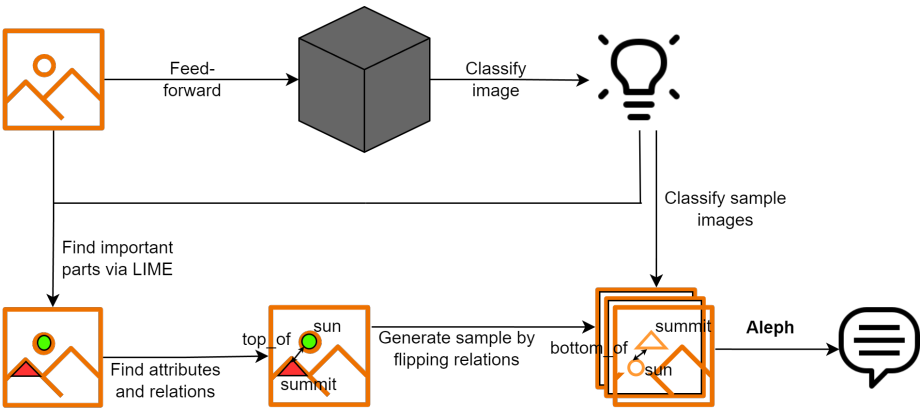


Figure 4.2: The LIME-Aleph workflow.

The following sections will go deeper into the process of the approach we dubbed LIME-Aleph:

As a running example, the simple Blocksworld domain described in Section 3.2 is employed. Although our approach is model-agnostic and can in general be applied to any classifier f, we kept in mind that typical models to classify images are CNNs. CNNs generally summarize information over regular grids over the image (see the explanation of filters and pooling layers in Section 2.1). This fact together with the blocksworld nature of our domain we used in the experiments led to the decision of altering the superpixel method used in LIME. In the original approaches, irregular chunks of pixels are constrained by finding sharp edges of a sudden change in color or intensity. Since we know that our blocksworld takes place in a regular grid, we simply set the boundaries of superpixels to the boundaries of that grid[2].

LIME can then locate the most important superpixels in our image instance x. Parts of high (positive and negative) correlation with the original class receive a high absolute value for the coefficients of the linear model. A hyper-parameter k defines how many of these most important parts we take into consideration for LIME-Aleph. All other parts are discarded. Note that we also keep parts with a high negative value, since these are indicators for critical positions in the image. When flipping such critical image parts, the classification can change and this interesting information will be included in the relational explanation. After this step, we are left with image regions and their locations (collected in data structure S).

The next step consists of automatically extracting symbolic information from the selection S. The user can control this step by giving a pool of attributes $\mathcal{A}$ and relations $\mathcal{R}$. Attributes can be anything that describe an image part like color, texture or even objects, given this information can be automatically extracted from the pixels. Also, relations can be manifold, for example spatial relations, differences in brightness or size etc. In general, any 2-ary relation can be used, as long as it is identifiable automatically and the inverse of it is

[2]This is of course a major restriction of possible domains as also discussed in Section 5.

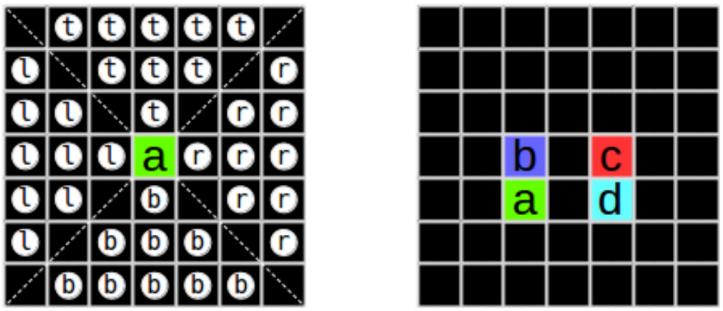


Figure 4.3: **Left**: Pivot block with position of blocks that are **r**ight, **l**eft, **t**op or **b**ottom of it. **Right**: Blocks in constellations `on(b,a)`, `on(c,d)`, `under(a,b)` and `under(d,c)`.

defined in image space. In the experiments described later, we focus on spatial relations, since the classes of our domain are defined by spatial relations.

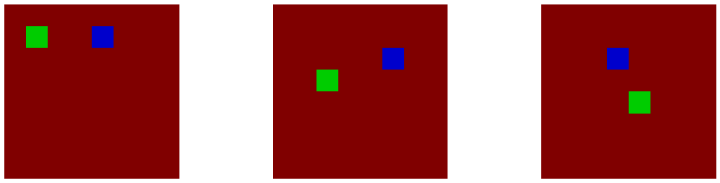
In our experiments, we only used the mean color as single attribute in $\mathcal{A}$. This attribute can take human-understandable color names as its values. These are found automatically by first identifying the mean color of an image part in RGB color space and then finding the nearest named color (using the Euclidean Distance) according to a pool of known named colors. These colors are extracted and symbolized for all image parts in S. A typical attribute for an image part $s \in S$ can for example be `green(s)`.

As spatial relations for our blocksworld domain, we populated $\mathcal{R}$ with `left_of`, `right_of`, `top_of`, `bottom_of`, `on` and `under`. The semantics of these relations for our domain can be seen in Figure 4.3. All relations that hold between all image parts in S are extracted and symbolized. So an example for an extracted relation would be `right_of(`p_1`,` p_2`)`, indicating that image part p_1 is located right of image part p_2.

LIME-Aleph aims at highlighting important relations that hold between important image parts. Therefore, just as with the original LIME, altered samples of the original image instance are created and the model behavior when classifying these samples is examined. To generate samples, we take each relation that was identified and invert it. This inversion is not only performed in the symbolic form, but also in image space. So e.g. for relation `right_of(`p_1`,` p_2`)`, we now have a new image z where the image parts p_1 and p_2 are flipped in the image and additionally the symbolic description of the image now features relation `right_of(`p_2`,` p_1`)`.

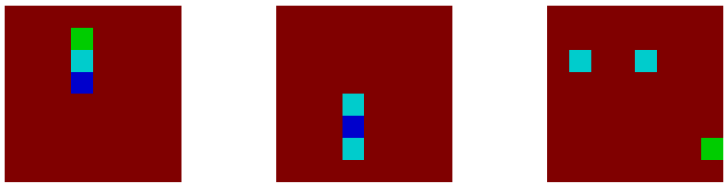
The next step consists of deciding whether the new symbolic description of the image should belong in the positive or negative set of examples for ILP. To accurately resemble the behavior of the original model f, we find the classification $f(z)$. If $f(z) = \oplus$, the image is put in the positive set, if $f(z) = \ominus$, it is put in the negative set. Finally, a first-order theory is induced by Aleph.

A first experiment used the simple concept "green block has to be left of blue block". Positive and negative instances can be seen in Figure 4.4. Parameter k was set to 3. For image (a) in Figure 4.4, Aleph generated the following perfect rule for the given concept:



(a) (b) (c)

Figure 4.4: Positive (a, b) and negative (c) examples for the concept "green left of blue" (from Rabold et al. (2020a)).



(a) (b) (c)

Figure 4.5: Positive (a) and negative (b, c) examples for concept "tower" (from Rabold et al. (2020a)).

```
concept(A) :- contains(B, A), has_color(B, green), contains(C,
              A), has_color(C, blue), left_of(B, C).
```

Although we allowed one extra background tile, Aleph abstracted from this irrelevant information.

For image (b) with the same hyper-parameters, we received the following rule, again perfectly resembling the concept:

```
concept(A) :- contains(B, A), has_color(B, blue), contains(C, A),
              has_color(C, green), left_of(C, B).
```

A second, more complex concept "tower" was introduced in a subsequent experiment. A tower is present, when there is a blue block as a foundation. Directly on that foundation, two blocks need to be stacked, where one is green and the other is cyan. The order of those two top blocks does not matter. Figure 4.5 shows examples of the tower concept.

For image (a) from the tower images, with $k = 3$, LIME-Aleph produces the following rule:

```
concept(A) :- contains(B, A), has_color(B, cyan), contains(C, A),
                                on(B, C).
```

When evaluating this rule, it is clear that the cyan block can never be the foundation. It has to be either the second or third block from below. Allowing one additional tile to be part of the explanation ($k = 4$) yielded this output:

```
concept(A) :- contains(B, A), has_color(B, cyan), contains(C, A),
              has_color(C, blue), top_of(B, C).
```

Both generated rules capture parts of the concept tower, but are too general. Considering the two rules in combination however gives a clearer picture. While the first rule manifests that the cyan block can not be the foundation, the second rule basically gives the color of the foundation with an unspecific statement of where the other blocks are. This experiment shows, that finding expedient hyper-parameters is still an open problem.

4.2.2 Model-specific Relational Explanation Generation

So far, approaches were presented, that do not necessarily need to know what exactly the type of the model is. These methods were mainly interested in showing the relation between input and output of image samples when putting them through a model. This section is concerned with the case, where we have an insight into the learned parameters of a model (model-specific explanations). This thesis focuses on neural networks, and especially CNNs, since these are the most common ML models in computer vision.

Keep in mind that, although the parameters in the model are now accessible, we are still dealing with a black-box. Even an expert will not be able to accurately predict the behavior of the model by merely looking at the learned weights. However, when using methods of Concept Embedding Analysis (see Section 2.3.2), it will be possible to extract information about visual concepts that is embedded in the network filters. When further finding the location of these visual concepts in an original image that was put through the model, we are able to also find relations that hold between concepts. Symbolizing this information enables us again to find a set of logic rules via ILP, explaining the behavior of the model.

It is important to keep in mind that, ultimately, we want to explain model behavior to humans that are directly working with the models. This can be end users or – in an earlier stage of development of the models – domain experts that already have a good idea of what are important constituents of Machine Learning decisions. Therefore, in publication Rabold et al. (2020b), an attribution method like LIME or Grad-CAM was completely omitted. Instead, experts can decide themselves which visual concepts are extracted.

As a simple relational domain, we created the Picasso dataset (see Section 3.3) consisting of images of human faces with either a normative relational structure of their facial features or a shuffled structure. Multiple standard image classification architectures were trained to tell whether the image shows a normative face (positive class) or a shuffled one (negative class). To locate the pixels corresponding to the semantic concepts of eye, nose and mouth, we trained per-pixel segmentation models, directly building upon the work of Fong and Vedaldi (2018) with minor adaptions (See Section 2.3.2). The original face dataset we used to build the Picasso set (FASSEG (Khan et al., 2015)) contained per-pixel masks for the facial features. These were used as ground-truth for the segmentation models to detect the location of eye, nose and mouth in new images, effectively generalizing the "expert" annotation to unseen images.

In contrast to the aforementioned approaches, we do not want to explain the decision for a single image, but for the behavior of the complete model (global explanation). This means that no particular image will stand out and we will not need to generate altered versions of it. Instead, we sample from a pool of readily available, unaltered test images to get an idea of how the model

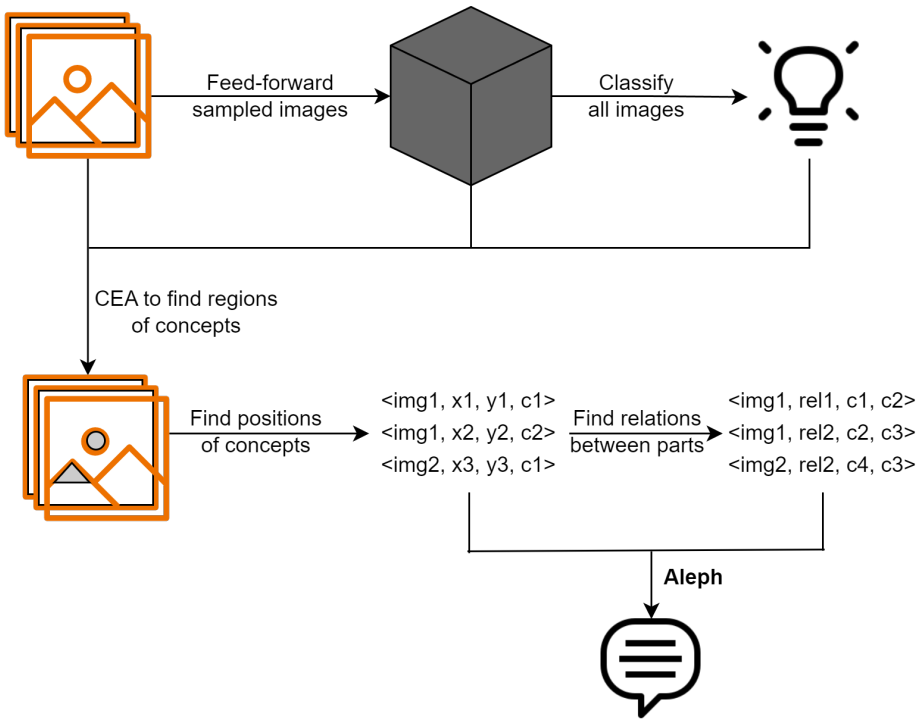


Figure 4.6: The model-specific workflow.

behaves. To stay close to the original model, the sampling favors images that are close to the decision boundary of the model. The proximity to the boundary is quantified by looking at the raw output of the last layer in the model and only taking images whose output is close to the class threshold. More on this in Section 4.3.

To find a relational explanation for the black-box models, we took a sample of 50 positive and 50 negative images closest to the decision boundary (See also Section 4.3). Per-pixel masks for the facial features were extracted using the described concept embedding analysis approach (See Section 2.3.2). The positions of the features were determined by taking the middle of the bounding box of a mask. For the two eyes, we took the two biggest "eye masks" and calculated their position separately. Spatial relations were found straight-forwardly given the positions as indicated in Figure 4.7. In accordance to our general explanation generation pipeline, the symbolized relational information together with a statement of what facial concept is located at a particular position was used as background knowledge for Aleph. As already described in the previous sections, when building the positive and negative examples for Aleph, we deviate from the ground-truth, but take the classification result of our original model. See Figure 4.6 for the workflow of this approach.

Table 4.1 summarizes the found explanations for the concept "normative face" for three test runs with three common neural network architectures. The rules all share a common part where nose and mouth are ground. The distinct rule parts feature spatial relations describing a further un-ground facial part,

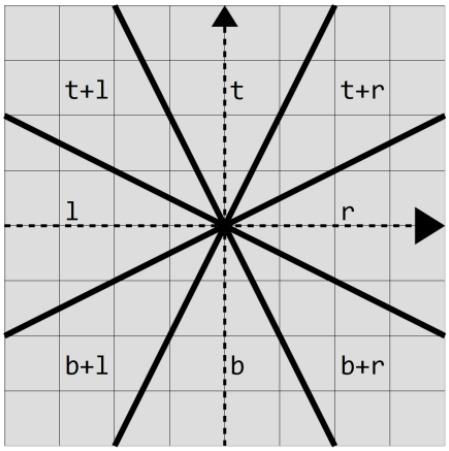


Figure 4.7: Regions of positions of concepts in order to be in relation **t**op of, **b**ottom of, **l**eft of, **r**ight of or a combination of relations (Reference object in the middle).

that is either above the ground parts or to the side of the nose. This can only be the eyes and thus, by construction, the rules correctly describe a normative face.

Table 4.1: Induced explanations for different architectures examined in Rabold et al. (2020b). The rules are of the common form `face(F) :- contains(F,A), isa(A,nose), contains(F,B), isa(B,mouth), distinctPart.`. The fidelity scores **Accuracy** and **F1** w.r.t the original model output will be explained in more detail in Section 4.3.

Arch.	**Accuracy**	**F1**	**Distinct Rule Part**
VGG16	99.60%	99.60%	`top_of(A,B), contains(F,C), top_of(C,A).`
AlexNet	99.05%	99.04%	`contains(F,C), left_of(C,A), top_of(C,B), top_of(C,A).`
ResNeXt	99.75%	99.75%	`top_of(A,B), contains(F,C), top_of(C,A).`

4.3 Fidelity of Explanations

This thesis is concerned with finding post-hoc explanations. That means an already trained ML model is available and the training process is finished. While it might be possible to inspect the parameters of the trained model, no efforts are made to retrain parts of the model to make it more interpretable. Instead, the previous chapters have shown ways to generate a relational surrogate for the black-box model. While a point was already made that ILP is helpful to generate expressive, human-interpretable surrogates, one question was left out: What efforts are made to have a surrogate accurately resemble the behavior of

the original model? And how can we quantify this? In other words: How can we mitigate and quantify the fidelity-interpretability trade-off? This section will give an answer to this and, thus, to research question **SQ2**.

It is in the nature of explanations that such a trade-off exists, since each simplification of network behavior for the sake of explanation is inherently a loss of complexity and thus, a loss of generalization power. To accurately explain the behavior of a model, we need to find *important* constituents of the problem setting. An important source of information which parts of an image input are relevant are visual attribution methods like LIME or Grad-CAM. Attribution methods are a reliable and well-studied way of examining (image) input w.r.t. an ML model and generating a heatmap of the image parts most important to a model decision (see e.g. Alicioglu and Sun (2022)). Insignificant image parts can be discarded. This ultimately helps in making the surrogate generation process more efficient.

When it comes to explaining the decision for a single image instance (local explanation), the previous sections have shown that it must be possible to find images similar to the one in question. That way, the behavior of the model in the vicinity of the pivot image can be examined. This can either be done by directly altering the image space of the image and creating altered versions of it (as in LIME or LIME-Aleph). Alternatively, when a set of images (like a test set) is available, we need to find a way of sampling images that are similar to the one pivot image. This vicinity set of images is needed, since the ILP system needs to build a set of positive and negative symbolic examples to generate an explanatory rule set.

Similarity metrics can be used to compare the pivot instance with other instances. Here, I want to highlight two possible similarity metrics; one on the image-level and one that can be used when the model parameters are inspectable.

A typical image-level similarity metric is the pixel-wise Euclidean metric. Equation 4.1 gives the formula for it, where I_P, I_Q are ordered lists of indices in images P, Q. For the ancient graves domain (see Section 3.1) we had to deal with sparse black-and white images. The Euclidean metric sometimes performs poorly when images are rather semantically similar, but just translated a few pixels. For this domain especially, sufficient results were achieved by first downsampling the images considerably and then calculating the Euclidean metric on the now denser pixel information.

$$\sqrt{\sum_{[x,y]\in I_P,[u,v]\in I_Q} (P[x,y]-Q[u,v])^2} \tag{4.1}$$

This metric can then e.g. be used in controlling the Aleph induction process. In its default implementation, Aleph uses the coverage metric c to guide the search for suitable clauses. This metric simply calculates coverage($Clause$) = $p - n$ where p is the number of positive examples covered by $Clause$ and n is the number of negative examples, this clause covers. In Rabold et al. (2018), we incorporated similarity of image instances into a new cost function *cost*. The rationale behind this function is to guide Aleph's search to favor many covered positive examples with small distance to the example to be explained over many covered negative examples with small distance. Equation 4.2 states this function where $c(e) = \frac{1}{(1+d(e))^2}$, $d(e)$ is the distance of example e to the

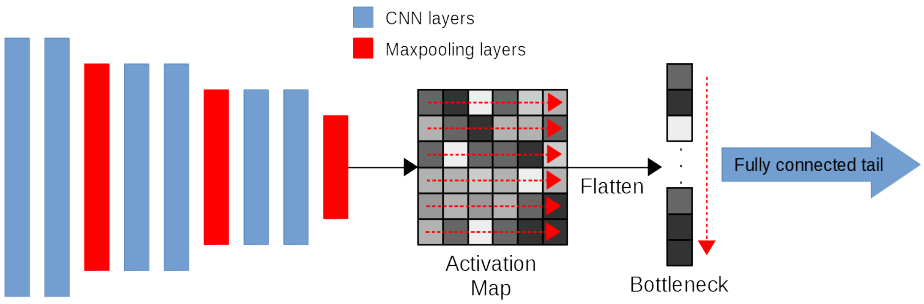


Figure 4.8: The location of the bottleneck within a typical CNN.

to be explained example, and E^+ and E^- are the sets of positive and negative examples respectively.

$$
\begin{aligned}
cost(Clause) = \\
&\sum_{e \in E^+} \begin{cases} -c(e), & \text{if } e \text{ gets covered by the clause} \\ 0, & \text{otherwise} \end{cases} \\
&+ \\
&\sum_{e \in E^-} \begin{cases} 0, & \text{if } e \text{ does not get covered by the clause} \\ c(e), & \text{otherwise} \end{cases}
\end{aligned} \tag{4.2}
$$

When inspection of parameters of a CNN model is possible, another metric of finding images to populate the vicinity set can be used. The idea is finding images that are not only similar in image space, but also finding images, the original model "thinks" are similar. We want to explain behaviors of ML models after all. First, we summarize each image into one vector. This is done by letting the model classify the image. But instead of taking the output of the last layer, we extract the flattened vector output of the last feature extraction layer we call the bottleneck layer (see Figure 4.8). This vector can be seen as a summary of the image with its most important features. Now, to compare images and finding suitable candidates for the vicinity set, we take the cosine similarity (see Equation 4.3 for two vectors $\vec{a}$, $\vec{b}$) of each image's bottleneck vector to the pivot image bottleneck vector[3].

$$
\frac{\vec{a} \cdot \vec{b}}{||\vec{a}||\ ||\vec{b}||} \tag{4.3}
$$

To explain the model behavior in general, without a particular image decision in mind (global explanation), we made the assumption, that image instances closer to the decision boundary of the original model are more interesting than instances further away. These instances seem to be on the brim of being assigned another class, given its attributes would be slightly different. This idea was sparked by considerations of counterfactual explanations and near miss examples (see e.g. Mothilal et al. (2020) and Rabold et al. (2022)).

[3]This method was used e.g. in Rabold (2022).

A simple way of seeing which images are near the decision boundary is examining the network output. When considering a 2-class classification task, images close to the classification threshold (e.g. a softmax output of 0.5 with 0 and 1 being the two extremes of classification) can be seen as being close to a class switch and therefore most interesting to explaining the model. When finding k instances for the vicinity set, we can simply find the k instances having an output closest to the classification threshold. This approach was taken for the image sampling in Rabold et al. (2020b).

Ultimately, the FOL rules have to resemble the behavior of the original black-box model. Especially in visual relational domains, a qualitative inspection of the generated rules can shed light on their plausibility for the domain in question. Rabold et al. (2020b) and Finzel et al. (2024) use the domain of the Picasso faces further described in Section 3.3. By construction of the dataset, it is clear how the image constituents have to be arranged in order to show a normative face. The generated rules for all neural architectures reproduce this structure.

Additionally, a quantitative analysis of the generated rules was performed in Rabold et al. (2020b) as well as in Finzel et al. (2024). The general idea is taken from the fidelity metric introduced in Guidotti et al. (2018b), which uses a metric similar to the accuracy metric. However we are not comparing the black-box output of images against their ground truth labels. Instead, we handle the black-box output as the new ground truth and compare it with the classification of the symbolic representation of the images when letting the relational surrogate model infer their label. That way, we receive a quantitative evaluation of fidelity of the generated explanation rules. Equation 4.4 states this fidelity measure, where E is the set of instances, $f(e) \in \{0, 1\}$ is the binary output of the original model f for e, $g(e) \in \{0, 1\}$ is the binary output of the explanation model g for e and $\mathbb{I} \in \{0, 1\}$ is the indicator function of an event). Table 4.1 lists the fidelity values for the three architectures used in Rabold et al. (2020b) when finding explanations in the Picasso Domain (see Section 3.3). In Finzel et al. (2024), similar high explainer fidelities are reported. When fine-tuning a VGG16 network (Simonyan and Zisserman, 2015) with training data from the Picasso dataset the fidelity is reported with 0.9860. For the test data we receive 0.9980 (with 1.0 being the highest possible fidelity). Finzel et al. (2024) actually reports high fidelity values for ILP surrogate models trained on a large variety of domains supporting the quantifiable effectiveness of an explanation generation approach based on visual attribution and ILP. Details can be found in the respective paper.

$$\frac{1}{|E|} \sum_{e \in E} \mathbb{I}(f(e) = g(e)) \tag{4.4}$$

4.4 Usefulness of Explanations

So far, we focused on the pipeline of how human-interpretable explanations can be generated to accurately mimic a black-box model. This, however, is only part of the answer of how humans can benefit from relational explanations that go beyond simple visual attributions. In the sense of Michie's strong and ultra-strong criteria discussed earlier, this section will showcase and discuss efforts in

creating explanations that adapt to the needs of individual human users working in a variety of domains and, thus, give an answer to research question **SQ3**.

In an effort to understand how the domain affects the explanation need, and which types of relational explanations are helpful for which explanation situations, an empirical study was conducted in Rabold et al. (2022). Participants were introduced into a purely abstract relational domain (Family domain, see Section 3.4) and a visual relational domain (Arches domain, see Section 3.5).

Participants were then presented with different explanation modalities or pairings of such. The modalities were:

- **General Rule (R)**, a relational rule, converted into natural language, describing a particular concept in the domain.
- **Example (E)**, a particular example belonging to the concept (positive example).
- **Near Miss (N)**, an example not in the concept (negative example), but structurally similar to a positive example.
- **Far Miss (F)**, a negative example not similar to a positive example.

A near miss example is an example not part of the concept in question, because just a few attributes are not in favor of the concept. An example from the Family domain given in Rabold et al. (2022) is given for the concept of `grandfather`:

$$\text{male}(A) \wedge \text{parent}(A, C) \wedge \text{parent}(C, B) \rightarrow \text{grandfather}(A, B)$$

A grandfather `A` is thus defined as a male parent of a parent of a given person `B`. A near miss could be a grandmother, whose concept can be defined like this:

$$\text{female}(A) \wedge \text{parent}(A, C) \wedge \text{parent}(C, B) \rightarrow \text{grandmother}(A, B)$$

Note, that as described in Rabold et al. (2022), a near miss *explanation* is a minimally changed concept rule, instantiated with a negative example. So e.g. for Ian, the grandfather of Kate (see Section 3.4), a possible near miss explanation would be:

$$\text{female}(\text{jodie}) \wedge \text{parent}(\text{jodie}, \text{tom}) \wedge \text{parent}(\text{tom}, \text{kate}) \rightarrow \text{grandfather}(\text{jodie}, \text{kate})$$

When presented with selected pairs of pairings of the explanation modalities explained above, an evaluation of relative frequencies (in brackets) of pairings yielded the following preference ranking for the Family domain:

RE (0.32) > **RN** (0.21) > **EN** (0.19) > **EF** (0.13) > **RF** (0.13) > **NF** (0.02)

For the Arches domain, the ranking looked like this:

EN (0.27) > **RE** (0.25) > **EF** (0.21) > **RN** (0.14) > **RF** (0.08) > **NF** (0.04)

Table 4.2: Mean ratings for the helpfulness of the explanation modalities (given in the first row) for the two domains (adapted from Rabold et al. (2022)). The highest mean rating for each purpose is in bold font.

	(R)ule	**(E)xample**	**(N)ear Miss**	**(F)ar Miss**
Family				
general	**4.97**	4.52	2.93	2.19
example	4.14	**4.70**	2.49	2.37
exclusion	2.95	2.62	**4.30**	3.67
Arches				
general	4.45	**4.70**	2.70	2.36
example	**4.56**	4.27	2.74	2.38
exclusion	3.25	2.73	**3.95**	3.82

Interestingly, for the abstract relational domain, the concept rule together with an example in that concept ranked highest, with a combination of the rule and an example that just falls shy of being in the concepts comes in at second place. For the visual relational domain, a combination of example and near miss is the top preference with a very close second place being again the concept rule and an example. Note, that in the Arches domain, the (near/far) examples were all drawings. The results indicate, that the combination of verbal relational explanations and a (visual) example is helpful for humans to understand results of ML systems. Evidence from psychology (Mayer and Sims, 1994) further backs this up, indicating that humans in an educational environment can benefit from a multi-modal explanation of a given task. Receiving an explanation for the behavior of an ML model is arguably an educational environment. Especially the combination of visual and verbal explanations gave rise to more creative solutions given by study participants in subsequent transfer tasks.

A last part of the study asked the participants to rate on a five-point scale for the two domains how helpful an explanation modality was to understand

- the **general** concept,
- a particular **example** instance for the concept,
- what is *not* in the concept (**exclusion)**.

The mean ratings over all participants is given in Table 4.2.

As expected, the near miss explanation was rated most helpful in both domains, to learn which examples are excluded from the concept. For the Family domain, the highest helpful ratings for understanding the general concept and a particular example are on par with the respective explanation modalities. Interestingly for the visual domain, in understanding the general concept, it was more helpful to see a visualization of an instance belonging to the concept. This is another indicator for the helpfulness of a combination of visual and verbal explanations in visual relational domains. For more information on the approaches and statistical methods involved in receiving these results, please refer to Rabold et al. (2022).

There needs to be a close connection between accurate ML and considerate practitioners. Effective models might be good at generalizing fast over training

data. However, when decisions are wrong, human experts in the problem domain can use their expertise to scrutinize them and use their well established common sense to solve conflicts or even correct models. In the field of Interactive ML, typically, model corrections are performed by a human-in-the-loop (Holzinger, 2016). In the focus of this thesis, when dealing with post-hoc explanations, corrections do not take place. However, working closely with a human can contribute to generating explanations, that speak the language of the respective experts. When relational explanations use predicates whose semantics is known to practitioners, the behavior of ML models can be grasped faster and errors can be recognized easier.

In Rabold (2022) I show how experts can be actively involved in the explanation generation pipeline. Instead of having to search for datasets pre-annotated with semantic concepts, experts can determine which visual concepts will be incorporated in explanations simply by pinpointing a few examples of concepts in images. Additionally, they can also decide how much time they want to spend for selecting them, since the approach already works for just a few annotated concepts. The approach first asks the user to provide a set of names of semantic concepts. In the running example in Rabold (2022), I use a network trained to discriminate between images showing dogs and cats as well as a set of exemplary cat images (see Section 3.6). The aim is to generate a relational explanation of the behavior of the model in the vicinity of a pivotal cat image. Therefore, a sample S of $N = 20$ similar images is selected (Similarity is evaluated by comparing the bottleneck layer outputs as described in Section 4.3). Next, Grad-CAM (see Section 2.3.1) is applied to all samples $s \in S$ on a particular convolution layer L. The $K = 10$ most important image patches in s (according to Grad-CAM) together with the corresponding vectors V_s^* across all feature maps on the output of L are aggregated. Specifically, if $[i, j]$ is element of the top K positions in the activation map stack $V_s = o(L)$ consisting of n activation maps[4], then $o_{[1:n]}(L)[i, j] \in V_s^*$ (see again Figure 2.3).

The pool of semantic concepts C is set to *eye*, *ear* and *whiskers*. Then, out of the $N = 20$ sampled cat images, only $n = 4$ images were selected to keep the additional work a user has to do to get an explanation manageable. For each of the n images, the user labeled the K most important image patches to the best of their knowledge with concepts from the concept pool. Patches where the user did not find an appropriate concept were discarded.

The next step is to generalize the user annotation to the larger sample set of N images to create an explanation that accurately resembles the black-box model behavior. In Rabold (2022), metric learning was used to achieve that. Supervised metric learning (Bellet et al., 2015; Kulis et al., 2013) uses a set of discretely labeled vectors to learn a vector transformation, effectively bringing vectors with the same label closer together according to a simple pre-defined distance metric D like Euclidean or Cosine (and also pushing vectors with different labels away from clusters of the same label). The Mahalanobis distance (see Equation 4.5) is often used to act as a learnable distance D' for the original labeled vectors.

$$D'(x, y) = \sqrt{(x - y)^\top M(x - y)} \tag{4.5}$$

This pseudo-metric closely resembles the Euclidean distance metric, but with

[4]See Section 2.1 for the description of notation for CNNs.

a matrix M, that does not necessarily have to be the identity matrix. In the Large Margin Nearest Neighbor (LMNN) method for metric learning, M is optimized by a loss to encorporate as many "target" nearest neighbors (points of the same label w.r.t. to a test point) when using the Mahalanobis distance while excluding points with different labels. The learned matrix can then also be used to transform a vector space, such that similar points according to the labeling cluster together with simpler metrics (as described above).

This method is often a preparation step for cluster algorithms like k-means, where the underlying distance metric of the samples is complex and/or simply unknown. This can be used to our advantage: We treat the concept annotations of the user as labels for the metric learning and find a transformation θ with all feature vectors V_s^* in all annotated samples as input for metric learning. The vector space is thus transformed in a way that features which the user deemed to be encoding similar concepts (like all belonging to patches of an *eye*) cluster together. We can now effectively also use θ to transform all feature vectors at all important image patches in all of the N sampled images to get the set of transformed vectors V_θ^*.

In a next step, a clustering algorithm with V_θ^* finds $|C|$ cluster centers. With these centers, we can start to locate concepts on all images $s \in S$. We first apply θ on all locations of activation map V_s (not only the important ones). For all $c \in C$, we then find the location of the vector $v \in V_s$ that has the highest cosine similarity (see Equation 4.3) to the cluster center corresponding to c. Table 4.3 compares the average of the maximum cosine similarities of all previously unlabeled images with metric learning either switched on or off. There is always a (slight) improvement for an experimental run with metric learning. This shows that the method is helpful in finding concept locations in unlabeled images with a better fit to human-evaluated similarity, even when the number of labeled images is rather small ($n = 4$).

The location of the vector with the highest similarity is then upsampled to find the location of the concept on the original image. Figure 4.9 shows cat images with overlayed heatmaps indicating the regions with highest similarity to concept-defining vectors of *eye*, *ear* and *whiskers*.

Table 4.3: Average of the maximums of cosine similarities of previously unlabeled images for each concept. Bold values indicate an improvement in similarity. In parentheses, the standard deviation is given.

	Eye	**Ear**	**Whiskers**
Without metric learning	0.31 (0.10)	0.33 (0.11)	0.25 (0.06)
With metric learning	**0.35** (0.12)	**0.34** (0.12)	**0.29** (0.07)

As in the approaches described above, the concept locations can then be used to find relations between concepts and generate symbolic relational explanations. The complete process we dubbed SymMetric is visualized in Figure 4.10.

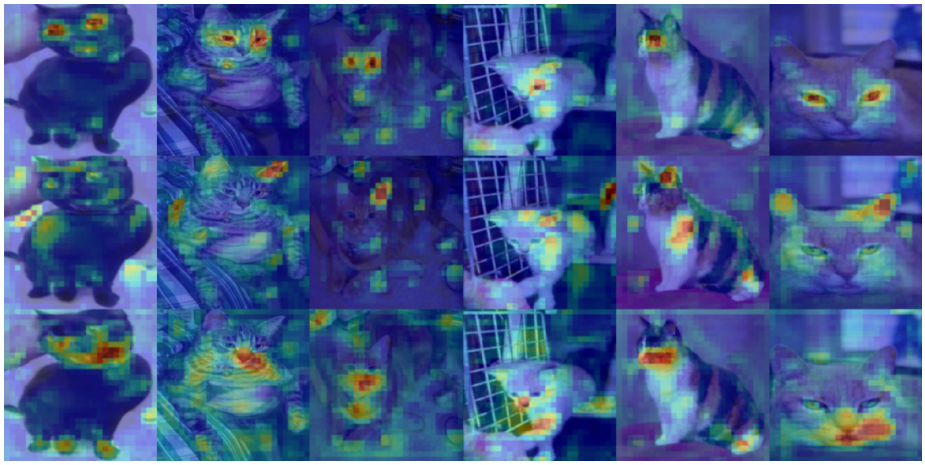


Figure 4.9: Previously unlabeled cat images (from the Microsoft® Research Dogs vs. Cats dataset (see Section 3.6)) overlaid with heatmaps indicating regions of high cosine similarity of the underlying transformed feature vector when compared to the concept-defining vector (the center vector of a respective concept cluster). The concepts are (from top to bottom): *eye*, *ear*, *whiskers*.

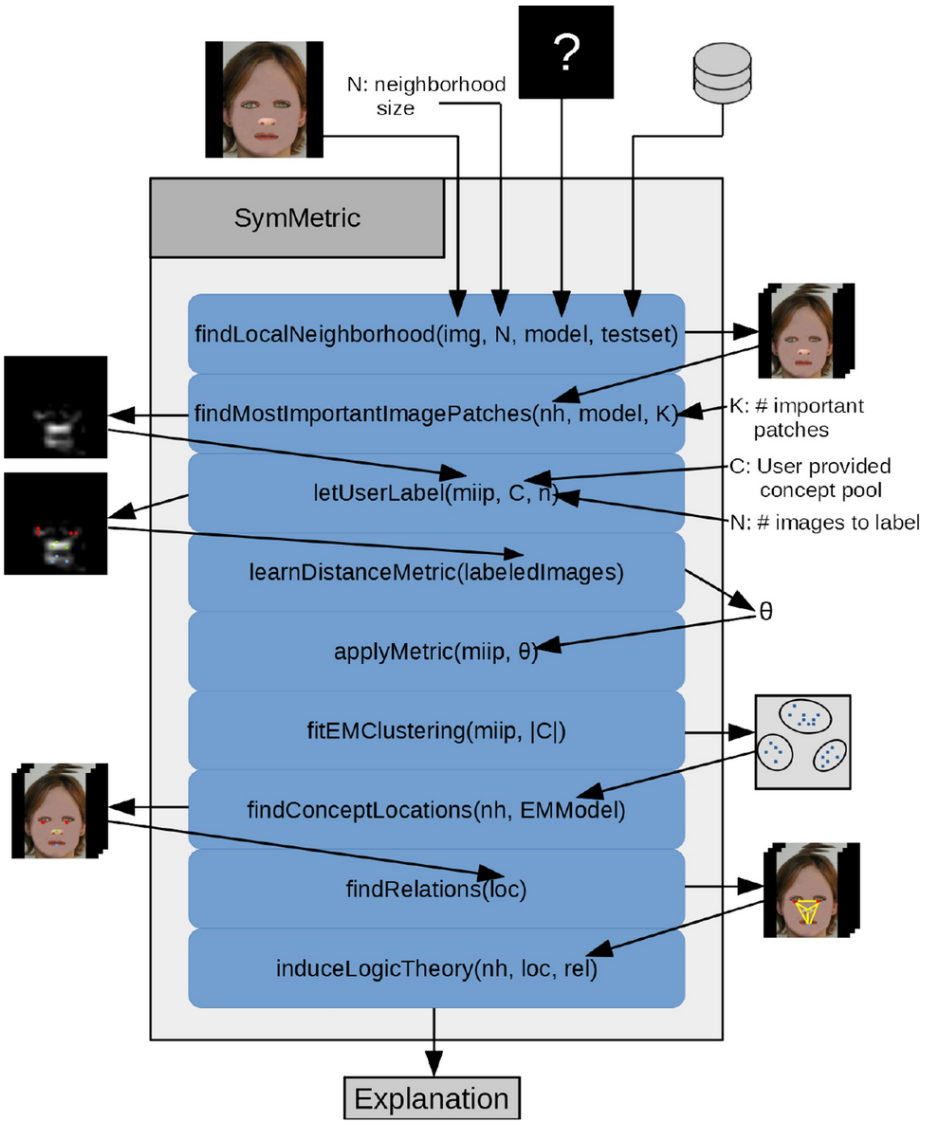


Figure 4.10: Workflow of the SymMetric approach. The user just needs to annotate a small sample in the vicinity of the to be explained instance. Metric learning then generalizes this annotation to a larger sample, symbolizes it and finally generates a relational explanation (Figure adapted from Rabold (2022) with example images from the Picasso domain (see Section 3.3)).

5
Conclusion and Outlook

This thesis described approaches on how to generate expressive, faithful and useful relational explanations for automatic decisions in visual relational domains. First, a general and re-usable pipeline of how to obtain important information and symbolizing them for ILP was laid out and discussed. This proof-of-concept was then refined to be usable for several common problem instantiations: In situations where the black-box model's parameters are not inspectable (model-agnostic), the focus has to lie on extracting important features directly from the image. Visual attribution methods find superpixels that are important for a decision and automatic color matching can be used to describe objects symbolically. When parameters are inspectable, we can make use of concept embedding analysis to give meaning to feature vectors and to localize particular semantic concepts. The pool of which concepts to use can either be given by general-purpose per-pixel-labeled datasets or by humans directly, effectively making it possible to explain in almost any domain.

For local explanation approaches, the bottleneck inherently is the availability of additional data. In order to build a sample around the to be explained image, you either need to alter the one image or rely on e.g. test data where you can find a vicinity with the help of similarity metrics. When altering one image from a relational domain is the only possibility, you need to find ways of finding alternative versions where relations are different in order to be able to see what relations the original model deems to be important (by examining the classification results of these altered versions and inducing a logic theory). The LIME-Aleph approach is thus not generally applicable to any domain, since it is not always possible to exchange two parts in image space without producing holes. This is a limitation that has to be investigated in future work.

To enforce fidelity on relational surrogate models, one can act on several parts in the pipeline. When generating an explanation for a single image's decision, it is useful to generate a sample of images close to the one in question. This sample will act as the examples (classified by the original black-box) used in the induction step of Aleph. Simple similarity measures like Euclidean or cosine, either applied on the images or an intermediate network output can be used here. These similarity values can either be used to build the sample in the first place, or as guidance term in the search loss for Aleph. Again, when working in a model-agnostic setting, one has to be aware of the limitations of trivial similarity metrics like Euclidean or cosine. While they might work for structured, non-sparse images, there is actually a problem for sparse images

like for example in the Ancient Graves domain. When comparing structurally identical images, where one image is just translated by a few pixels, the metrics actually will show a very low similarity, since they work pixel-wise and the identical pixels would not overlap anymore. This can however be leveraged by using metrics like the structural similarity index measure (Wang et al., 2004) or, as in this work, by downsampling the image to "summarize" it.

When the general behavior of a model has to be explained, it showed advantageous to sample images that were being classified close to the network decision boundary. The faithfulness of a final relational explanation in form of a logic theory (w.r.t. the original black-box) can then also be post-hoc quantified by calculating an accuracy-based fidelity measure. The resulting explanations in the experiments discussed in this work always showed high fidelity measures or were plausible considering the construction rules of the concept in the respective domain. We have to be careful however to not interpret the final softmax network output as a probability, just because it is limited to the range between zero and one. Research in model uncertainty inherent in neural networks (Gawlikowski et al., 2023) hints at other methods of measuring how uncertain a network was in its decision and thus, how close the instance was to the decision boundary. These methods however need some additional computational effort, and so this work relies on the easily obtainable rough estimate of proximity to the decision surface.

For an explanation to be useful for humans working with automatic decision systems, the generation process has to adapt to the domain. Our empirical studies have shown that there is a difference in preference of explanation modalities when working in an abstract relational domain in contrast to a visual relational one. Also, the preferred modality for explanation purposes (like grasping the general concept or finding what is excluded from the concept) is domain dependent. When incorporating humans in the explanation generation, they can also actively guide the process. By just a few hand-crafted annotations, it is possible to generalize these annotations to other non-annotated images in order to build a sample in the vicinity of a pivotal image. That way, experts in specialized fields are able to build relational explanations based on fine-grained constituents, without relying on pre-labeled datasets.

Since we are trying to find explanations for humans, it would be preferable in the future, to always evaluate approaches in an empirical study. Especially when aiming at implementing an explanation module in a manufacturing process involving many experts, there needs to be an additional feedback loop to check, what expectations users have, if they adopt the new technology and ultimately, if they trust it.

6
Papers and Contributions

The following papers and their results are part of this thesis. My contributions are stated below the respective paper.

1. **Johannes Rabold**, Michael Siebers, Ute Schmid. "Explaining Black-box Classifiers with ILP – Empowering LIME with Aleph to Approximate Non-linear Decisions with Relational Rules". *Inductive Logic Programming – 28th International Conference, ILP*. Springer International Publishing, 2018. `https://doi.org/10.1007/978-3-319-99960-9_7`

   I contributed text to explain the LIME algorithm and to describe our combination of LIME with the ILP framework Aleph. I also co-developed the adapted clause evaluation function for Aleph that fit the needs for our verbal explanation generation approach. The experiments of the paper were conducted by me and I was the author of the respective texts and the statement of the experimental results. The generator for producing example images for our Graves domain was developed by me. I also co-authored the conclusion.

2. **Johannes Rabold**, Hannah Deininger, Michael Siebers, Ute Schmid. "Enriching Visual with Verbal Explanations for Relational Concepts – Combining LIME with Aleph". *Machine Learning and Knowledge Discovery in Databases – International Workshops of ECML/PKDD 2019*. Springer International Publishing, 2020.
   `https://doi.org/10.1007/978-3-030-43823-4_16`

   I was the author of text explaining the LIME algorithm. I also developed the approach of how to extract relevant symbolic information (parts and their relations) from our domain images and how to create local verbal explanations. Also I contributed the text for these sections. The experiments were conducted by me and the accompanying text as well as the statement of results were written by me. The proposed combination of visual and verbal explanations was co-developed by me and the text was authored by me. I co-authored the conclusion.

3. **Johannes Rabold**, Gesina Schwalbe, Ute Schmid. "Expressive Explanations of DNNs by Combining Concept Analysis with ILP". *KI 2020: Advances in Artificial Intelligence: 43rd German Conference on AI*. Springer International Publishing, 2020.
   https://doi.org/10.1007/978-3-030-58285-2_11

   I wrote the theoretical background section on ILP. The conceptual idea of using internal embeddings to explain a neural network by a set of symbolic rules described in this paper was developed by me. I also wrote the text explaining this approach and developed an algorithm for using the results of concept embedding analysis to extract symbolic background knowledge. The selection process of sampling instances close to the decision boundary of the network was mainly developed and authored by me. The experiments on verbal explanation generation as well as the statement of the results were developed and authored by me.

4. **Johannes Rabold**, Michael Siebers, Ute Schmid. "Generating Contrastive Explanations for Inductive Logic Programming based on a Near Miss Approach". *Machine Learning 111.5, p: 1799-1820*. Springer, 2022.
   https://doi.org/10.1007/s10994-021-06048-w

   The function of near miss examples was co-authored by me. I wrote the text for explaining the basic concepts in ILP. The definitions of near miss examples and explanations was co-developed and co-written by me. The algorithm as well as the accompanying proofs were co-developed by me. The experiments conducted for the Family and the Arches domain were executed by me and I wrote the text as well as the results for them. The empirical study was co-developed and executed by me and I co-authored the accompanying text. I also wrote parts of the conclusion.

5. **Johannes Rabold**. "A Neural-symbolic Approach for Explanation Generation based on Sub-concept Detection: An Application of Metric Learning for Low-time-budget Labeling". *KI - Künstliche Intelligenz 36.3, p: 225-235*. Springer, 2022.
   https://doi.org/10.1007/s13218-022-00771-9

   All text was written by me in single-authorship. I also developed the complete approach of generalizing human annotations to a bigger sample using metric learning for verbal explanation generation by myself. This includes the developed algorithm. The experiments and results were developed, executed and found by me.

6. Bettina Finzel, Patrick Hilme, **Johannes Rabold**, Ute Schmid. "When a Relation Tells More Than a Concept: Exploring and Evaluating Classifier Decisions with CoReX". *Machine Learning submission. Latest version can be found here:*
   https://doi.org/10.48550/arXiv.2405.01661

   I contributed to the conduction of the experiments by pre-processing a dataset of pathological image data and training neural network models on them. Additionally I co-developed and co-authored text concerning the Picasso dataset and the used altered accuracy measure to quantify fidelity of the generated surrogate explanation models.

Bibliography

A. M. Abdel-Zaher and A. M. Eldeib. Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46:139–144, 2016.

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. doi: 10.1109/ACCESS.2018.2870052.

G. Alicioglu and B. Sun. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics*, 102:502–520, 2022.

M. Bakator and D. Radosav. Deep learning and medical diagnosis: A review of literature. *Multimodal Technologies and Interaction*, 2(3):47, 2018. doi: 10.3390/mti2030047.

D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

A. Bellet, A. Habrard, and M. Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015. ISBN 978-3-031-00444-5. doi: 10.2200/S00626ED1V01Y201501AIM030.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

M. Campbell, A. J. H. Jr., and F. Hsu. Deep blue. *Artificial Intelligence*, 134 (1-2):57–83, 2002. doi: 10.1016/S0004-3702(01)00129-1.

W. F. Clocksin and C. S. Mellish. *Programming in PROLOG*. Springer Science & Business Media, 2003.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893. IEEE Computer Society, 2005. doi: 10.1109/CVPR.2005.177.

A. S. d'Avila Garcez, K. Broda, and D. M. Gabbay. *Neural-symbolic learning systems - Foundations and applications*. Perspectives in Neural Computing. Springer, 2002. ISBN 978-1-85233-512-0. doi: 10.1007/978-1-4471-0211-3.

A. S. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *IfCoLog Journal of Logics and their Applications (FLAP)*, 6(4):611–632, 2019.

L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

L. Deng and D. Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3-4):197–387, 2014. doi: 10.1561/2000000039.

T. G. Dietterich. Learning at the knowledge level. *Machine Learning*, 1:287–315, 1986.

Directorate-General for Communication. Rules for business and organisations. `https://commission.europa.eu/law/law-topic/data-protection/reform/rules-business-and-organisations_en`, 2023. [Online; accessed 21-September-2023].

R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018. doi: 10.1613/jair.5714.

B. Finzel, P. Hilme, J. Rabold, and U. Schmid. When a relation tells more than a concept: Exploring and evaluating classifier decisions with CoReX. 2024. doi: 10.48550/arXiv.2405.01661. URL `https://arxiv.org/abs/2405.01661`.

R. Fong and A. Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8730–8738, 2018.

J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77, 2023.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

H. Greenspan, B. van Ginneken, and R. M. Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016. doi: 10.1109/TMI.2016.2553401.

H. P. Grice. Logic and conversation. In *Speech Acts*, pages 41–58. Brill, 1975.

M. Gromowski, M. Siebers, and U. Schmid. A process framework for inducing and explaining datalog theories. *Advances in Data Analysis and Classification*, 14(4):821–835, 2020.

R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018a.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):1–42, 2018b.

D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. Xai—explainable artificial intelligence. *Science Robotics*, 4(37):eaay7120, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

J. A. Hendler, A. Tate, and M. Drummond. Ai planning: Systems and techniques. *AI Magazine*, 11(2):61–61, 1990.

A. Holzinger. Interactive machine learning for health informatics: When do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, 2016.

K. Khan, M. Mauro, and R. Leonardi. Multi-class semantic segmentation of faces. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 827–831. IEEE, 2015.

R. D. King, S. H. Muggleton, A. Srinivasan, and M. Sternberg. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences*, 93(1):438–442, 1996.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012.

B. Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computing*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. D. Raedt. Deepproblog: Neural probabilistic logic programming. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3753–3763, 2018.

S. Marya, R. Thukral, and C. Singh. Prosthetic replacement in femoral neck fracture in the elderly: Results and review of the literature. *Indian Journal of Orthopaedics*, 42(1):61, 2008.

R. E. Mayer and V. K. Sims. For whom is a picture worth a thousand words? Extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology*, 86(3):389, 1994.

D. L. Medin and M. M. Schaffer. Context theory of classification learning. *Psychological Review*, 85(3):207, 1978.

D. Michie. Machine learning in the next five years. In D. H. Sleeman, editor, *Proceedings of the Third European Working Session on Learning, EWSL 1988, Turing Institute, Glasgow, UK, October 3-5, 1988*, pages 107–122. Pitman Publishing, 1988.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

S. H. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994. doi: 10.1016/0743-1066(94)90035-3.

S. H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. R. Besold. Ultra-strong machine learning: comprehensibility of programs learned with ILP. *Machine Learning*, 107(7):1119–1140, 2018. doi: 10.1007/s10994-018-5707-3.

D. Müller, M. März, S. Scheele, and U. Schmid. An interactive explanatory AI system for industrial quality control. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 12580–12586. AAAI Press, 2022.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. doi: 10.1023/A:1022643204877.

J. Rabold. A neural-symbolic approach for explanation generation based on sub-concept detection: An application of metric learning for low-time-budget labeling. *KI - Künstliche Intelligenz*, 36(3):225–235, 2022. doi: 10.1007/s13218-022-00771-9.

J. Rabold, M. Siebers, and U. Schmid. Explaining black-box classifiers with ILP - empowering LIME with aleph to approximate non-linear decisions with relational rules. In F. Riguzzi, E. Bellodi, and R. Zese, editors, *Inductive Logic Programming - 28th International Conference, ILP 2018, Ferrara, Italy, September 2-4, 2018, Proceedings*, volume 11105 of *Lecture Notes in Computer Science*, pages 105–117. Springer, 2018. doi: 10.1007/978-3-319-99960-9_7.

J. Rabold, H. Deininger, M. Siebers, and U. Schmid. Enriching visual with verbal explanations for relational concepts - combining LIME with aleph. In P. Cellier and K. Driessens, editors, *Machine Learning and Knowledge Discovery in Databases - International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16-20, 2019, Proceedings, Part I*, volume 1167 of *Communications in Computer and Information Science*, pages 180–192. Springer, 2020a. doi: 10.1007/978-3-030-43823-4_16.

J. Rabold, G. Schwalbe, and U. Schmid. Expressive explanations of dnns by combining concept analysis with ILP. In U. Schmid, F. Klügl, and D. Wolter, editors, *KI 2020: Advances in Artificial Intelligence - 43rd German Conference on AI, Bamberg, Germany, September 21-25, 2020, Proceedings*, volume 12325 of *Lecture Notes in Computer Science*, pages 148–162. Springer, 2020b. doi: 10.1007/978-3-030-58285-2_11.

J. Rabold, M. Siebers, and U. Schmid. Generating contrastive explanations for inductive logic programming based on a near miss approach. *Machine Learning*, 111(5):1799–1820, 2022. doi: 10.1007/s10994-021-06048-w.

L. D. Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In M. M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2462–2467, 2007.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016. doi: 10.1145/2939672.2939778.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

E. Rich. Artificial intelligence, 1983.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

L. Schallner, J. Rabold, O. Scholz, and U. Schmid. Effect of superpixel aggregation on explanations in lime–a case study with biological data. In *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I*, pages 147–158. Springer, 2020.

G. Schwalbe and B. Finzel. A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, pages 1–59, 2023.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 618–626. IEEE Computer Society, 2017. doi: 10.1109/ICCV.2017.74.

L. Serafini and A. S. d'Avila Garcez. Logic tensor networks: Deep learning and logical reasoning from data and knowledge. In T. R. Besold, L. C. Lamb, L. Serafini, and W. Tabor, editors, *Proceedings of the 11th International Workshop on Neural-Symbolic Learning and Reasoning (NeSy'16) co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence (HLAI 2016), New York City, NY, USA, July 16-17, 2016*, volume 1768 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

M. Siebers and U. Schmid. Please delete that! Why should i? - Explaining learned irrelevance classifications of digital objects. *Künstliche Intelligenz*, 33 (1):35–44, 2019. doi: 10.1007/s13218-018-0565-5.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

A. Srinivasan. *The Aleph Manual*, 2007. `https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html` [Accessed: 13.08.2023].

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1355.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1-2):119–165, 1994. doi: 10.1016/0004-3702(94)90105-8.

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

D. S. Watson, J. Krutzinna, I. N. Bruce, C. E. Griffiths, I. B. McInnes, M. R. Barnes, and L. Floridi. Clinical applications of machine learning algorithms: beyond the black box. *The BMJ*, 364, 2019.

P. H. Winston. *Learning structural descriptions from examples*. PhD thesis, Massachusetts Institute of Technology, USA, 1970.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.