Applied Intelligence
https://doi.org/10.1007/s10489-024-05321-4

Concept logic trees: enabling user interaction for transparent image classification and human-in-the-loop learning

David M. Rodríguez[1,2] · Manuel P. Cuéllar[1] · Diego P. Morales[1]

Accepted: 4 February 2024

Abstract
Interpretable deep learning models are increasingly important in domains where transparent decision-making is required. In this field, the interaction of the user with the model can contribute to the interpretability of the model. In this research work, we present an innovative approach that combines soft decision trees, neural symbolic learning, and concept learning to create an image classification model that enhances interpretability and user interaction, control, and intervention. The key novelty of our method relies on the fusion of an interpretable architecture with neural symbolic learning, allowing the incorporation of expert knowledge and user interaction. Furthermore, our solution facilitates the inspection of the model through queries in the form of first-order logic predicates. Our main contribution is a human-in-the-loop model as a result of the fusion of neural symbolic learning and an interpretable architecture. We validate the effectiveness of our approach through comprehensive experimental results, demonstrating competitive performance on challenging datasets when compared to state-of-the-art solutions.

Keywords Soft decision trees · Concepts · XAI · Neural symbolic · Image classification · Human-in-the-loop

1 Introduction

Interpretable machine learning models are increasingly important in domains where transparent decision-making is required. Miller [26] introduced several considerations for implementing new interpretable AI models. The author emphasized that explanations are a form of knowledge transfer resulting from interaction. Many studies have focused on interpreting models based on black boxes or defining interpretable models. However, we believe that defining interpretable solutions enabling user interaction with the model is an understudied area. This study addresses that research gap by exploring the fusion of soft decision trees, neural symbolic learning, and concept learning. The objective is to develop an interpretable classification model enabling user intervention and incorporating expert knowledge. Our fusion proposal is based on the following arguments:

– Concept learning facilitates human intervention and interpretation [18]. Furthermore, it enables the use of neural symbolic learning and the definition of first-order logic predicates based on human-understandable concepts.
– Neural symbolic learning enables the definition of rules to articulate, intervene, and explore the model's decision-making process.
– If the user is not able to understand the decision-making process, the interaction with the model is not possible. The use of soft decision trees enables the user to understand the decision-making process. Additionally, the use of soft routing enables the integration with fuzzy logic. This allows users to define first-order logic rules for intervening in the routing process.

Our main contribution is a novel solution in the field of image classification, designed to inherently provide interpretability due to its transparent architecture. By integrating neural symbolic learning via Logic Tensor Networks, our model enables users to incorporate expert knowledge through the definition of first-order logic rules and predicates. Moreover,

✉ David M. Rodríguez
dmorales@correo.ugr.es ; david.morales@hattec.de

Manuel P. Cuéllar
manupc@ugr.es

Diego P. Morales
diegopm@ugr.es

1 University of Granada, 18071 Granada, Spain

2 HAT.tec Lilienthalstraße 15, 85579 Neubiberg, Germany

the proposed fusion of concept learning and neural symbolic learning enables the user to inspect the model by making queries based on different classes or combinations of concepts. A key contribution of our approach is the ability to impose constraints on the soft decision tree's routing process. These constraints, specified as first-order logic rules and predicates based on concept or class combinations, provide users with greater control over the decision-making process. Additionally, our proposed approach enables users to inspect the decision routes taken by the model through queries. All these features together introduce transparency and interpretability by providing insights into the model's decision-making process.

We evaluate our proposed approach on challenging datasets and compare its performance to state-of-the-art solutions, demonstrating competitive results. Furthermore, we discuss future research directions, including the potential of combining neural symbolic learning and soft decision trees in reinforcement learning domains.

The article is organized as follows: first, we provide an overview of related work in Section 2. Next, the proposed approach is presented in Section 3. Then, in Section 4 we describe the experiments and discuss the results. Finally, Section 5 presents future research directions and conclusions.

2 Related work

While the initial machine learning algorithms were transparent to the user and easily interpretable, in recent years, opaque decision systems like deep neural networks (DNNs) have become the "de facto" solution for many machine learning problems in different critical context fields such as medicine, defense or system safety [1, 6, 37]. However, solutions based on DNNs are considered 'black-box' machine learning models whose behavior can be hard to explain. Consequently, there has been a growing interest in explainable artificial intelligence (XAI). Post-hoc explanations, which involve interpreting methods after training the models are widely adopted approaches for explaining DNNs [1]. Some well-known post-hoc explanation techniques as Local Interpretable Model-Agnostic Explanations (LIME) [31] or Class Activation Mapping (CAM) [39] identify the specific regions of input features that the networks focus on when making predictions.

On the flip side, achieving transparent deep learning models is a primary objective of XAI and an actively researched area. Traditional machine learning models and algorithms like decision trees or k-NN offer interpretability and transparency but are outperformed by opaque models such as deep neural networks. Consequently, recent research has been dedicated to resolving this well-known trade-off between performance and explainability [10, 27, 32], aiming to define models that are inherently transparent and do not require post-hoc explanation techniques.

In this search for more explainable model architectures, some authors have aimed to fusion the transparency of decision trees and the power of deep learning methods. Decision trees are considered transparent models as following the decision paths enables humans to understand the rationale behind a prediction or classification [29]. However, as previously mentioned, decision trees do not generalize as well as neural networks. Kontschieder et al. [19] introduced Deep Neural Decision Forests, aiming to combine representation learning from deep learning with the divide-and-conquer principle of decision trees. They introduced a stochastic and differentiable decision tree called "neural decision tree". The proposed solution is an ensemble of these neural decision trees known as a decision forest. Wan et al. [36] presented their approach Neural-Backed Decision Trees (NBDT). They proposed a hierarchy-learning-based model where every node of a decision tree is formed by a neural network that makes low-level decisions. This approach induces hierarchies that can be used to explain the model's decision-making process. Frosst and Hinton [12] proposed distilling a neural network into a Soft-Decision-Tree (SDT). They described a method that utilizes a pre-trained neural network to train a soft decision tree using stochastic gradient descent and the predictions of the neural network as targets. Their model makes hierarchical decisions based on the learned filters and selects a particular static probability distribution over classes as the output.

Another interesting approach that has gained attention in the search for transparent models is concept-based explainability. Researchers exploring this approach aim to develop interpretable models by designing them to rely on concepts as the basis for their decision-making. Concepts, in this context, refer to high-level and semantically meaningful units of information such as color, texture or shape. The resonating process of the models can be interpreted by generating explanations that are based on those concepts [24].

One of the most known research articles in this field was published by [18], who presented concept bottleneck models (CBMs). They proposed to use a CNN as a concept extractor that maps raw inputs (x) to concepts (c). After that, a second model maps these concepts (c) to targets (y) performing the final classification. Other authors have proposed to train object detectors or segmentation models as concept extractors to localize object parts that are used as concepts. The final model solution combines those models with a classifier that bases its decision on the detected object parts [3, 6].

While many of these studies have concentrated on interpreting models with black-box characteristics or creating interpretable models, we consider that the exploration of interpretable solutions enabling user interaction remains a relatively underexplored research area. Miller [26] outlined several factors to consider during the implementation of novel interpretable AI models. The authors emphasized the nature of explanations as a form of knowledge transfer resulting from interactions. In this context, [18] proposed a concept-learning model that allows concept intervention. Mispredicted concepts can be corrected using expert knowledge, enabling users to refine and improve the model's predictions [18, 38].

With the aim of merging transparent architectures with concept learning, we investigated the fusion of concept bottleneck models and soft decision trees in our previous article [28]. However, although this fusion enabled the generation of explanations by analyzing the decision paths based on human-understandable concepts, the human interaction or intervention and the use of expert or background knowledge were limited.

Conventional neural networks and deep learning methods do not take domain or background knowledge into account [35]. In recent years, the use of symbolic approaches to avoid these limitations has been subject of study [2, 34, 35]. The inclusion of symbolic knowledge in the form of first-order logic constraints into the loss function during deep networks training is a promising approach, that has been shown to enhance the fairness of the machine learning system while also preserving its performance [34]. In this field, an interesting framework was presented by [2]. Their proposal Logic Tensor Networks (LTN) allows defining variables, and predicates, where the variables are grounded by tensors, and predicates can be grounded by any neural network. The power of this approach relies on the definition of relations among the predicates that can be established as logical rules. Those rules can be used to define metrics or loss functions. The application of this framework to different tasks such as logic reasoning [2], object detection [22], zero shot learning [25] or image segmentation [9] has been demonstrated.

In this article, we investigate the fusion of neural symbolic learning and concept learning with the aim of developing an interpretable deep learning model. Our proposed model combines a soft decision tree and a concept-based model to define an interpretable model that performs image classification basing its decision on human-understandable concepts. The final decision-making process is conducted by a soft decision tree that can be visualized and explored by the user. The use of neural symbolic learning enables human intervention, as an expert can explore the decision tree and improve it by using his knowledge to redefine the decision tree. This results in a learning cycle where the user can control and intervene in the training process. This learning cycle can be observed in the diagram presented in Fig. 1.

3 Methodology

In this section we present our proposed solution that corresponds to the learning cycle presented in Fig. 1. We propose the fusion of three approaches that have been shown to improve the transparency of deep learning models: neural symbolic learning, Soft Decision Trees and Concept Bottlenecks. Our study involves employing Logic Tensor Networks to train a CNN as a concept extractor, mapping images to concepts. A Soft Decision Tree serves as a predictor for final classification, utilizing the extracted concepts. Figure 2 illustrates the architecture diagram of our proposed model. The utilization of neural symbolic learning during the training phase allows for the inclusion of domain knowledge and provides a tool for interpreting results during testing and inference time.

3.1 Concept bottlenecks

In the field of concept learning, the image classification problem is usually formalized as follows [18]: Let us consider an input vector, denoted as $\mathbf{x} \in \mathbb{R}^d$, a target output, denoted as $y \in \mathbb{Y}$, and a concept vector, denoted as $\mathbf{c} \in [0, 1]^k$. The training dataset consists of samples of the form $[(\mathbf{x}_n, y_n, \mathbf{c}_n); n = 1...N]$.

We propose a model based on the concept bottlenecks presented in [18]. The proposed model takes the form $t(g(\mathbf{x}))$,

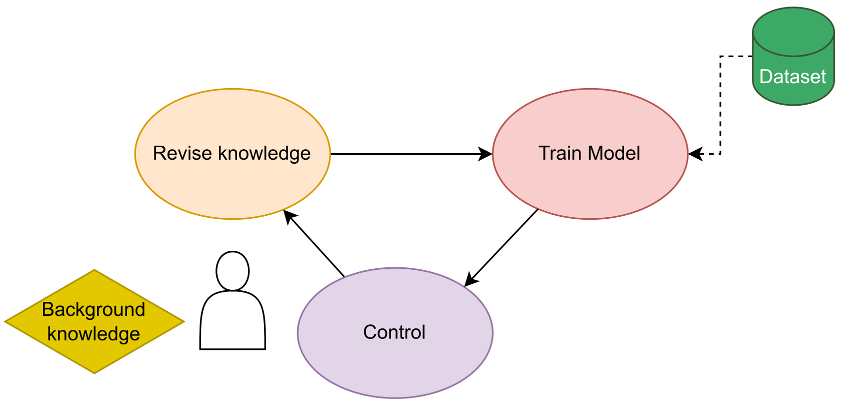

Fig. 1 Learning cycle. The use of neural symbolic learning enables human intervention. An expert can control the training process and intervene the model by revising the knowledge through the definition or re-definition of knowledge rules

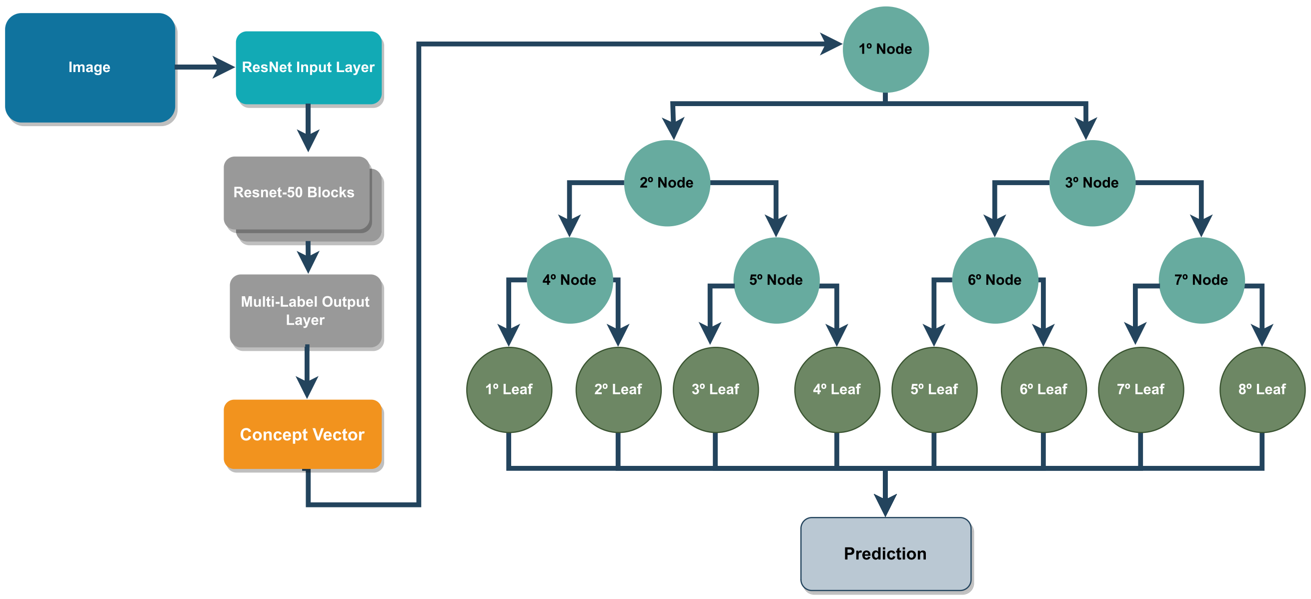

Fig. 2 Architecture diagram of the proposed solution. The concept extractor g (implemented by a Resnet-50) takes an image as input and outputs the concept vector. The binary soft decision tree t gets the concept vector as input and outputs the final prediction

where $g : \mathbb{R}^d \rightarrow [0, 1]^k$ maps the image from the input space to the concept space. A classification subnetwork $t : [0, 1]^k \rightarrow \mathbb{Y}$ maps $\mathbf{c} = g(\mathbf{x})$ from the concept space to the target space. In our case, this classification subnetwork is implemented by a soft decision tree. This model can be trained by combining two loss functions: A classical classification loss function $L_Y : \mathbb{Y} \times \mathbb{Y} \rightarrow \mathbb{R}^+$, which measures the discrepancy between the model's output $y' = t(g(\mathbf{x_i}))$ and the target output y_i for a given training sample $(\mathbf{x_i}, \mathbf{c}_i, y_i)$. A concept loss function $L_c : [0, 1]^k \times [0, 1]^k \rightarrow \mathbb{R}^+$ that measures the discrepancy between the output of the concept extractor $g(x_i)$ and the true concept vector $\mathbf{c}_i$. This loss function captures the dissimilarity between the predicted and actual concept vectors. By optimizing these two loss functions, the proposed model learns to associate the input features with the relevant concepts and make predictions based on the learned concept-target relationships. The authors proposed three ways of training the concept bottlenecks:

– Independent bottleneck: the two models t and g are trained independently. That is, g is trained on the training set $[(\mathbf{x}_n, y_n, \mathbf{c}_n); n = 1...N]$ minimizing $\sum_{n=1}^{N} L_c(g(\mathbf{x}_n); \mathbf{c}_n)$ while t is trained on the corresponding concepts subset by minimizing $\sum_{n=1}^{N} L_Y(t(\mathbf{c}_n); y_n)$
– Sequential bottleneck: t is trained on the output of g. That is, t minimizes $\sum_{n=1}^{N} L_Y(t(g(\mathbf{x}_n)); y_n)$. The concept extractor g is trained as before.
– Joint bottleneck: g and t are trained jointly by minimizing the combined loss function $\sum_{n=1}^{N} L_Y(t(g(\mathbf{x}_n)); y_n) + \delta \sum_{n=1}^{N} L_c(g(\mathbf{x}_n); \mathbf{c}_n)$. The hyperparameter $\delta > 0$ controls the trade-off between the two losses.

3.2 Soft decision trees

Traditional decision trees employ deterministic routing, where each sample is directed to exactly one path at each node. However, this deterministic routing introduces discontinuities in the loss function, making classical decision trees unsuitable for gradient descent-based optimization algorithms [15]. Consequently, classical decision trees cannot be trained using such algorithms. To overcome this limitation, we propose a model based on the binary soft decision tree presented in [12]. Unlike classical decision trees, soft decision trees employ probabilistic routing (or soft routing) instead of deterministic routing. This soft routing technique ensures that the loss function remains continuous, enabling the use of gradient descent-based optimization methods [12, 15]. Like classical trees, soft decision trees are formed by nodes and leaves. Given an input feature $\mathbf{x}$, the probability of taking the right branch at node i is calculated as:

$$p_i(\mathbf{x}) = \theta(\mathbf{x}\mathbf{w}_i + \mathbf{b}_i) \tag{1}$$

where θ represents the logistic sigmoid function, and $\mathbf{w}_i$ and $\mathbf{b}_i$ are the learned parameters. The probability of routing to the left branch is $1 - p_i(\mathbf{x})$. Each leaf node l generates a probability distribution over the output classes. This is done by applying the softmax function on the learned parameters ϕ^l associated with the corresponding node l:

$$Q^l = \text{softmax}(\phi^l) \tag{2}$$

To train the tree, the following loss function is defined:

$$L_T(\mathbf{x}) = -\log(\sum_{l \in L} P^l(\mathbf{x}) \sum_{k \in Y} y_k \log Q_k^l) \tag{3}$$

where Y represents the set of possible labels, k is the index of the label, and y_k denotes the target probability of $\mathbf{x}$ belonging to class k (either 0 or 1). $P^l(\mathbf{x})$ corresponds to the probability of arriving at leaf node l given the input $\mathbf{x}$, that is

$$P^l(\mathbf{x}) = \prod_N p_i(\mathbf{x})^{\mathbf{1}[l \swarrow i]}(1 - p_i(\mathbf{x}))^{\mathbf{1}[i \searrow l]} \tag{4}$$

Here, the indicator function $\mathbf{1}$ evaluates to 1 if the condition holds, and 0 otherwise. The notation $[l \swarrow i]$ (and $[i \searrow l]$) indicates that leaf l belongs to the left (or right) subtree of node i. The output of the model is the distribution associated with the leaf having the maximum path probability. Additionally, a penalty term is incorporated to ensure balanced utilization of both the left and right subtrees, as mentioned in [12].

3.3 Logic tensor networks

Logic Tensor Networks is a neural-symbolic framework that enables the use of first-order fuzzy logic in combination with deep learning neural networks. By defining variables, constants, predicates and rules in a so-called knowledge base, the learning problem can be seen as an optimization problem, consisting on maximizing the satisfiability of the formulas defined in the knowledge base. As an example, consider a basic binary classification problem where we have samples $x \in X$ that can belong to class A or to B. Furthermore, we know that for any sample, if a given feature f_i is greater than 0, that sample belongs to class B. We can then define the knowledge base as follows

$$K = \{\forall x_a P(x_a), \forall x_b \neg P(x_b), \forall x : f_i > 0 \implies \neg P(x)\}$$

where the notation x_a represents samples belonging to class A and P is a predicate that outputs the probability of belonging to class A. To benefit from deep learning techniques, the predicate P can be grounded with a neural network with weights ϕ. To train a neural network in this environment, a loss function can be defined in the following way:

$$L_{Sym} = 1 - Sat_K(x, \phi) \tag{5}$$

where Sat is the satisfiability of the formulas defined in the knowledge base K, for a model with trainable weights ϕ. In our proposed solution the predicates involved in the rules defined by the user in the knowledge base are grounded on the model described before (see Fig. 2). This way we are able to train the model using the knowledge rules.

The rules and axioms defined can also be used to inspect the model and to interpret its decisions and behaviour [2, 9] on inference time. In this research work, we explore both approaches.

3.4 Proposed training approach

In this section, we describe the proposed training approach. This solution we describe allows us to train the model proposed above (shown in Fig. 2) according to the learning cycle introduced in Fig. 1. For this purpose, we combine the loss functions presented before in the Sections 3.1, 3.2 and 3.3. The equation for the proposed approach is presented in (6):

$$\sum_{n=1}^{N}(\alpha L_Y(t(g(\mathbf{x}_n)); y_n) + \beta L_{Sym}(\mathbf{x}_n, \phi) + \delta L_c(g(\mathbf{x}_n); \mathbf{c}_n)) \tag{6}$$

where the hyperparameters $\alpha, \beta, \delta > 0$ control the trade-off between the losses. In Fig. 3 a diagram of the training process is presented. The proposed model is fed with the images and outputs the concept and class predictions as inference results. The loss calculator gets those outputs and the rules defined in the knowledge base as inputs and calculates the training loss according to (6). The user can observe and control the model. Based on this observation, he can add knowledge in the form of first-order logic rules to the knowledge base or modify the already existing knowledge. This training process implements the learning cycle previously discussed in Fig. 1. Furthermore, we explore the combination of neural symbolic learning with each of the three possible concept-bottleneck models described in Section 3.1. The proposed solution explained above follows the Joint approach. For the independent and sequential approaches the training of the concept extractor would remain the same as explained in Section 3.1. The classifier training would be done by minimizing the (7):

$$\sum_{n=1}^{N}(\alpha L_Y(t(\mathbf{c}_n); y_n) + \beta L_{Sym}(\mathbf{x}_n, \phi)) \tag{7}$$

Please note that for the sequential approaches $\mathbf{c}_n = g(\mathbf{x}_n)$ while for the independent approaches, $\mathbf{c}_n$ would be the corresponding concept label, according to the original ideas presented in Section 3.1.

4 Experimental setup, evaluation and results

This section provides an overview of the two datasets utilized for evaluating the proposed method. Subsequently, we outline the implementation details and describe the experiments

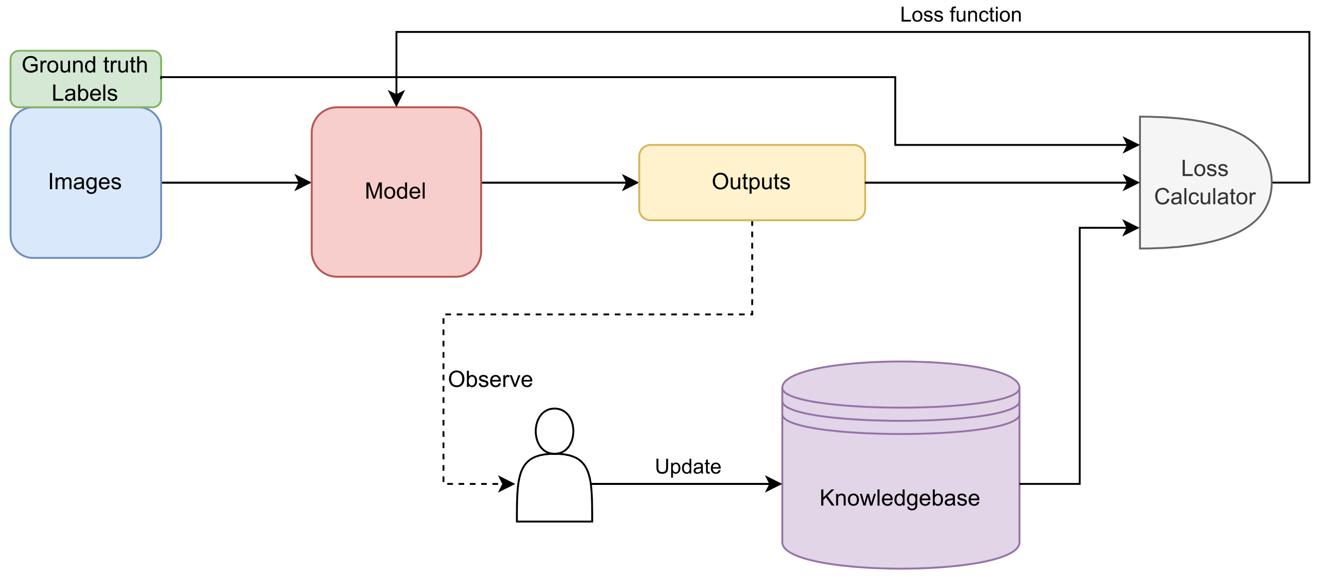

Fig. 3 Diagram of the proposed training process. The user can redefine the knowledge base by adding or modifying rules. These rules are used to compute the symbolic loss L_{Sym} as explained in Section 3.3. The final loss function is calculated aggregating all losses L_Y, L_c and L_{Sym} as explained in (6)

conducted, along with the evaluation metrics employed. Lastly, we present and analyze the results obtained.

4.1 Datasets

We evaluated the proposed methods on two datasets: the Semantic PASCAL-Part dataset [8] and the MonuMAI dataset [21].

The MonuMAI dataset [21] consists of over 1500 images of monuments belonging to four architectural styles: Gothic, Hispanic-Muslim, Renaissance and Baroque. The dataset has been expertly annotated, with human experts providing labels for monument style classification and key architectural element detection. Additionally, labels for fifteen key architectural element types (i.e. flat arch, pointed arch, porthole...) were also generated. The classification and analysis of those key elements can be used to explain the decision of a classifier as done in [3, 21].

The PASCAL VOC 2010 dataset [11] is a well-known image dataset comprising 20 object classes. Supplementary part-based annotations were provided in the PASCAL-Part dataset [5]. We evaluate the proposed approach on a curated version of the PASCAL-Part dataset presented in [6]. The authors aggrouped some similar categories in order to reduce the number of object part categories. Additionally, the images were selected so that only one main object class per image is present (classical image classification problem). The result is an image dataset containing over 1400 images belonging to 20 categories (i.e Person, TV, Train, etc.). Furthermore, the images are part-annotated on more than 40 different elements (i.e Leg, Body, Wheel,...). This dataset has already been explored on concept-based or part-based research articles [3, 6].

4.2 Implementation details

The proposed method was implemented on Pytorch, and the code is available for download.[1] The LTN-pytorch framework was used for the neural-symbolic setup. As the backbone for the concept extractor we use a Resnet-50 [16]. The soft decision tree is based on the model described in [12]. The code is based on the implementation provided in [7] and adapted to be integrated with the LTN framework. The depth parameter is set to 4 after a preliminary analysis. We used the Adam optimization algorithm [17] for all networks. The training scripts for the concept bottlenecks are based on the scripts provided by the authors of the original article [18].

4.3 Predicates and rules

We defined three different types of rules and their associated predicates:

– Basic class rules and predicates: one rule per class, they are of the form

$$\forall x_{gothic} P_{gothic}(x_{gothic})$$

. In this case, the rule specifies that for all samples of the class “gothic” (x_{gothic}) the predicate “is gothic”

[1] https://github.com/DavidMrd/LogicConceptSoftTrees

(P_{gothic}) should be true. We implemented the predicate as described in [2] for multi-class single-label problems. One of the advantages of this implementation is that the use of a softmax function ensures that no rules of the form

$$\forall x_{gothic} \neg P_{baroque}(x_{gothic})$$

are needed.

– Knowledge rules and predicates: They are of the form $\forall x : x_{pointed\_arch} \Rightarrow P_{gothic}(x)$. In this case, the rule specifies that all samples containing the concept "pointed arch" should be classified with the class label "gothic". These rules formed the knowledge bases presented in Figs. 4 and 5. This knowledge is extracted from the descriptions provided by the authors of the datasets [8, 21] and also from our own previous inspections of the datasets. In order to propagate gradients only over the corresponding subsets (in the example the subset of samples which verify the condition of "containing a pointed arch") we used guarded quantifiers [2] for the implementation. The predicates can be implemented the same way as above, with the only difference of using the concept labels instead of the class labels.
– Path rules and predicates: We can define two types of rules depending on the type of labels that they are based on:
  – based on class labels: $\forall x_{gothic} P_{path_1}(x_{gothic})$. In this example, the rule specifies that all samples belonging to the class "gothic" should follow the path $path_1$
  – based on concept labels: $\forall x : x_{pointed\_arch} \Rightarrow P_{path_2}(x_{gothic})$. In this example, the rule specifies that all samples containing a "pointed arch" should follow the decision path $path_2$.

  In these rules, $path_i$ is a vector of size n_{brach} the number of branches. The predicates P_{path_i} output the probability of going through the selected branches. Every $path$ can be a hole path from the top to one leaf or just a partial path (i.e. probability of visiting one specific node). To use these rules we had to adapt the soft decision tree so that we can get the probability of following a specific path as output. These rules allow the user to specify decision paths based on the classes (based on class labels) or based on the presence or absence of certain elements (based on concept labels).

Rule	If/Condition	Then/Conclusion
1	$PointedArch$	$Gothic$
2	$ConopialArch$	$Gothic$
3	$HorseshoeArch$	$Hispanic$
4	$LobedArch$	$Hispanic$
5	$TrilobedArch$	$Gothic$
6	$SalomonicColumn$	$Baroque$
7	$AdoveladoLintel$	$Hispanic$
8	$CurvedPediment$	$Hispanic$
9	$BullseyeWindow$	$\neg Hispanic \wedge \neg Gothic$
10	$GothicPinnacle$	$Gothic$
11	$Serliana$	$Renaissance \vee Baroque$
12	$SegmentalArch$	$Baroque \vee Renaissance$

Fig. 4 Knowledge base for the MonuMAI dataset. This knowledge base contains all knowledge rules for the MonuMAI dataset. See Section 4.3

4.4 Experiments and results

In this section, we introduce the experiments carried out to validate the proposed methods (see Section 3) on the datasets presented in Section 4.1. We kept the splits in training and test sets that were proposed in [3, 6].

4.5 Preliminary experiments

In this section we present some preliminary experiments that we carried out during the construction of the final solution. They have the character of an ablation study, as we tested the addition of some components to the final model.

4.5.1 Soft decision tree

In order to justify the use of a soft decision tree in the proposed solution, we performed the following experiment. We compared a soft-decision-tree-based concept bottleneck with a concept bottleneck based on a neural network classifier. The first model corresponds to the model of the proposed solution (without including the neural symbolic learning). For the second model, we used a multilayer perceptron (3-layers) as a classification subnet. We kept the Resnet-50 as concept extractor for all models. To make a fair comparison, we used the same extracted concepts for the independent and the sequential approaches (where the classifier is trained offline). We present the results in Table 1. That is the reason why the C-Acc for those two approaches is the same for all models. The Joint-Tree model gets the best results, achieving almost 2 points accuracy more than the second-best model on the main task. The independent and the sequential tree models perform slightly better than the corresponding baseline models.

4.5.2 Class rules and multi-class cross-entropy

In this subsection, we analyze the use of class rules and the optimization of their satisfiability to train the model compared to the classical approach based on a multi-class cross-entropy loss function. The use of class rules would allow us to go for a "pure" satisfiability optimization solution (pure LTN-Solution), while the use of a multi-class cross-entropy (CCE) approach would mean a fusion of training

Fig. 5 Knowledge base for the PASCAL dataset. This knowledge base contains all knowledge rules for the PASCAL dataset. See Section 4.3

Rule	If/Condition	Then/Conclusion
1	$AnimalWing \vee Beak$	$Bird$
2	$Stern \vee Engine \vee ArtifactWing$	$Aeroplane$
3	$Locomotive \vee Coach$	$Train$
4	$ChainWheel$	$Bicycle$
5	$Hoof$	$Horse$
6	Cap	$Bottle$
7	$Body$	$Bottle \vee Aeroplane$
8	$Ebrow \vee Foot \vee Arm \vee Hair \vee Hand \vee Mouth$	$Person$
9	$LicensePlate \vee Door \vee Bodywork \vee Mirror \vee Window$	$Car \vee Bus$
10	$Diningtable$	$Diningtable$
11	$Pot \vee Plant$	$Pottedplant$
12	$Saddle \vee Handlebar$	$Bicycle \vee Motorbike$
13	$Sofa$	$Sofa$
14	$Boat$	$Boat$
15	$Horn$	$Sheep \vee Cow$
16	$Chair$	$Chair$
17	$Screen$	$Tvmonitor$

approaches (satisfiability for the knowledge rules and classical approach for the classes). The results are presented in Table 2. Please note that these results correspond to models where no knowledge in form of attribute rules is present.

For the Independent and Sequential models the results are pretty similar. However, we found that the training of the sequential model based on a pure symbolic learning approach was difficult and even if we trained the model during much more epoches than the Joint-N model, the model did have many travels to converge and the finally Y-Acc got was much lower than the one got by the model using multi-class cross-entropy. After these results, we decided to use the classical approach based on multi-class cross-entropy, as the pure LTN approach does not bring any advantages and it needs more epochs to converge. Please note that these models do not incorporate attribute logics. The results for the proposed models are presented in the next section.

4.6 Results

In this section, we report and analyze the results obtained for the proposed approach on the two datasets.

In Table 3 we present the results obtained for the different methods on the proposed datasets. We evaluate how each proposed approach performs for two different tasks: concept extraction and final classification. Using the annotation presented in Section 3, given a trained concept extractor g and a trained tree t, we evaluate the classification task by computing the accuracy (Y-ACC) of the proposed bottleneck $t \circ g$, that is

$$\mathrm{Y-Acc} = Acc(y, y') \tag{8}$$

where y is the target, this is the given annotation label for the sample $\mathbf{x}$ and $y' = t(g(\mathbf{x}))$ is the final prediction of the proposed model. To evaluate how the concept extractor g performs, we compute the concept accuracy $C - Acc$, that is

$$\mathrm{C-Acc} = Acc(\mathbf{c}, \mathbf{c}') \tag{9}$$

where $\mathbf{c}$ is the vector representing the annotated concepts for a given sample $\mathbf{x}$ and $\mathbf{c}' = g(\mathbf{x})$ is the prediction of g for the sample $\mathbf{x}$. Furthermore for the neural symbolic approaches we compute the mean satisfiability Sat for the corresponding rules defined in the knowledge base K, given the model with trainable weights ϕ.

$$Sat = Sat_K(\mathbf{x}, \phi) \tag{10}$$

We repeated every experiment three times and present the mean results in Table 3. As baseline methods, we use the approaches presented in [28]. Note that the baseline methods have the same architecture as the proposed solutions, as described in Sections 3.4 and 4. The difference relays on the integration with neural symbolic learning via Logic Tensor Network and the incorporation of first-order logic into the training. We use the letter "N" to notate the proposed models that were trained using neural symbolic learning. "Ind," "Seq" and "Joint" represent the three different concept bottlenecks: Independent, Sequential, and Joint (see Section 3.1).

It can be observed that the Independent model and the Sequential model performed very similarly on both datasets. Please note that the C-Acc for those two approaches is the

Table 1 Results of the proposed previous experiments comparing the use of soft-decision-tree based models to multilayer-perceptron baseline models on the MonuMAI dataset for the already presented metrics

MonuMAI			
	Ind-Tree	Seq-Tree	Joint-Tree
Y-Acc	92.74	92.85	**97.69**
C-Acc	97.62	97.62	**97.64**
	Ind-Baseline	Seq-Baseline	Joint-Baseline
Y-Acc	92.34	92.41	95.93
C-Acc	97.62	97.62	92.79

Best results for each evaluation measurement are in bold

Table 2 Results of the proposed experiments to analyze the use of class rules (LTN models) compared to a classical approach (CCE models) on the MonuMAI dataset for the already presented metrics

MonuMAI						
	Ind-LTN	Ind-CCE	Seq-LTN	Seq-CCE	Joint-LTN	Joint-CCE
Y-Acc	92.1	92.74	92.21	92.85	78.22	**97.69**
C-Acc	97.62	97.62	97.62	97.62	96.92	**97.64**

Best results for each evaluation measurement are in bold

same since the same concept extractor g is used for both models, and only t is different. Please refer to Section 3 for more details.

The neural symbolic approaches achieve competitive performance compared to the baselines, despite the imposed constraints. This enhances the explainability of the models without creating a significant trade-off in their accuracy. The neural symbolic approaches offer the user the possibility of understanding the model decision process through the defined rules. In the table, we present the mean satisfiability for the rules, but for the user would it be also possible to know the satisfiability of each rule. This is translated into a good trade-off between explainability and performance. For example, the satisfiability per rule for the Seq-N on the PASCAL dataset is [88.91, 88.18, 80.24, 98.70, 99.64, 95.69, 90.04, 76.99, 89.28, 97.12, 85.56, 98.50, 90.61, 92.47, 95.19, 82.48, 81.85], where the rules are in the same order as presented in Table 5. In this way, we can see that the rule with the lowest satisfiability is rule number 8. This is not surprising as it is the only rule that involves more than 5 attributes. Inspecting the rule for the labels, we see that the satisfiability of the rule in the training set is of 99.99. So we can be sure that the rule is well-defined, so we should work on improving the behaviour of the model for these attributes and for the corresponding class, maybe giving it more importance (i.e. using balance weights during training).

In the case of the Joint approaches, which performed best for the baseline, we would like to note that we encountered particularly challenging training those models, as it required optimizing three loss functions simultaneously: the concept loss, the classification loss, and the symbolic loss. (see Section 3). For the MonuMAI dataset, in order to attain comparable accuracy to the baseline, the results for rule satisfiability decline compared to other approaches. By optimizing the parameters in the corresponding loss equation (see Section 3.1) the Joint model could be forced to pay more attention to the concepts, to the final prediction, or to the knowledge base. For the Pascal dataset, the results are really promising, although the concept accuracy is a bit lower than for other models. For the PASCAL dataset, our the Joint-N approach outperforms the corresponding baseline. We can see that not only the final Y-Acc is higher, but also that the performance of the concept extractor is higher than in the baseline. The model benefits from the knowledge base and the joint training approach, getting also the highest satisfiability, 91.55 (2 points higher than for the sequential approach).

4.7 Use case: user intervention

In this section, we present two use cases for the two studied datasets, in which the user modifies the behavior of the decision tree using logical rules. We first present a use case for the Pascal dataset, where our goal is to group the classes into three categories (animal, transportation, and indoor object) and force the tree to distinguish between these three categories at the initial nodes. To achieve this, we define the logical variables Animal, IndoorObj, and Transport, which represent the classes belonging to these categories. We define these variables based on attributes. For example, we define the Animal variable based on the presence of the following attributes: Torso, Tail, Neck, Eye, Leg, Beak, AnimalWing,

Table 3 Results of the proposed experiments on both datasets for the already presented metrics

MonuMAI						
	Baseline-Ind	Ind-N	Baseline-Seq	Seq-N	Baseline-Joint	Joint-N
Y-Acc	92.74	91.00	92.85	92.03	**97.69**	95.71
C-Acc	97.62	97.62	97.62	97.62	**97.64**	95.35
Sat	–	92.29	–	**94.01**	–	90.67
PASCAL						
	Baseline-Ind	Ind-N	Baseline-Seq	Seq-N	Baseline-Joint	Joint-N
Y-Acc	82.56	81.20	82.8	81.00	85.57	**89.25**
C-Acc	**97.21**	**97.21**	**97.21**	**97.21**	94.64	96.84
Sat	–	83.15	–	89.42	–	**91.65**

Best results for each evaluation measurement are in bold.

Head, and Ear. Based on these variables, we can define the following predicates:

- $\forall x_{\text{Animal}} \rightarrow P_{\text{path}}(x_{\text{Animal}}, l_{\text{path\_LTree}})$
- $\forall x_{\text{Transport}} \rightarrow P_{\text{path}}(x_{\text{Transport}}, l_{\text{path\_RTree}})$
- $\forall x_{\text{IndoorObj}} \rightarrow P_{\text{path}}(x_{\text{IndoorObj}}, l_{\text{path\_RTree}})$
- $\neg x_{\text{Animal}} \rightarrow \neg P_{\text{path}}(x_{\text{Animal}}, l_{\text{path\_RTree}})$
- $\neg x_{\text{Transport}} \rightarrow \neg P_{\text{path}}(x_{\text{Transport}}, l_{\text{path\_LTree}})$
- $\neg x_{\text{IndoorObj}} \rightarrow \neg P_{\text{path}}(x_{\text{IndoorObj}}, l_{\text{path\_LTree}})$

Observe that these predicates are defined as explained in Section 4.3 (see Path rules and predicates). In this way, we indicate the decision paths that should be taken based on these categories, so that the tree first groups the classes into categories before making the final class decision. The constant "$l_{\text{path\_LTree}}$" is defined as visiting the first branch (or left branch) of the tree (same for "$l_{\text{path\_RTree}}$"). For the MonuMAI dataset, we explore the option of defining rules based on the classes instead of basing the rules on the attributes as before. The rules are defined in the form:

- $\forall x_{\text{Hispanic}} \rightarrow P_{\text{path}}(x_{\text{Hispanic}}, l_{\text{path\_Hispanic}})$

We add one rule per class. For the class Renaissance, we define the corresponding path as the one ending on the last leaf node (number 8 starting from the left). For class Hispanic we define the path as going throw the left subtree and for the Renaissance class as going through the right subtree. For Baroque, we only add the constraint of not taking the path defined for Baroque.

We add these rules the corresponding knowledge base (keeping the rules used in the first experiments) and train the sequential models from scratch. We choose the sequential approach for this experiment because its results serve as a reference not only for itself but also as a minimum benchmark for the joint model. Take into account that the sequential model can be seen as taking $\delta \rightarrow \infty$ in the equation that defines the loss function for the joint model (see Section 3.1) [18]. We compute a separate satisfiability "Sat Path" for the new rules. We present the results in Table 4.

We can observe that these rules assist the user in defining the model's reasoning mechanism, with a low impact on the model's accuracy. In fact, in the case of MonuMAI, we can even observe that the intervened model achieves better results than the Sequential-N model. This should not come as a surprise since if the user is familiar with the task and can define rules that improve the model's decision process and help it in its task, it would enhance its performance. Furthermore, allowing the user to modify the model's reasoning process also opens the door to performing subtasks, as in the previous cases where we are implicitly conducting classification into meta-classes.

Table 4 Results of the proposed experiments on both datasets for the already presented metrics

MonuMAI			
	Baseline	Sequential-N	Sequential-N-Intervented
Y-Acc	**92.85**	92.03	92.77
Sat	–	94.01	93.24
Sat Path	–	49.35	79.12
PASCAL			
	Baseline	Sequential-N	Sequential-N-Intervented
Y-Acc	**82.80**	81.00	80.61
Sat	–	89.42	87.22
Sat Path	–	41.75	81.92

Best results for each evaluation measurement are in bold

4.8 Compare to the state-of-the-art

In this section, we compare our models and results to five state of the art approaches that were introduced above in Section 2. Three of the models are transparent models (Greybox [3], EXPLANet [6]) and [28] and the other two models are opaque models (DeiT-B [3, 33] and MonuNet [21]). MonuNet is an ad-hoc solution for monument-style classification, which is why results are not available for the PASCAL dataset. The results are presented in Table 5. Note that [28] is the method that we used as baseline in the sections above.

On the PASCAL dataset, our approach achieves higher accuracy than the explainable state-of-the-art models [3, 28, 33]. On the MonuMAI dataset, we achieve competitive results despite of the constraints added to the model.

Our proposal is explainable not only due to its architecture (as it is a soft decision tree that can be visualized as shown in [28]) as most of the other transparent approaches but also because it defines a knowledge base that allows the user to comprehend the model's behavior through rules, enabling them to even modify the decision-making process. Among the related works presented, the only proposal that also makes use of a knowledge base is [3], although they do not impose restrictions on the model during training; instead, they use the knowledge base as a training set, similar to how it is done in the independent model. However, this approach does not allow them to employ first-order logic or define additional rules based on other factors, such as nodes to visit. The inability to define first-order logic-based rules around this knowledge base is a limitation that our proposal overcomes. We also present the results for the Sequential intervented model, demonstrating that this proposed solution where the user is able to modify the decision process using first-order logic-based rules also achieves competitive results. Note that it is the only approach where the user has this option. This demonstrates that the proposed neural symbolic approach allows the users to define constraints not only based on the

Table 5 Results compared to the state of the art

Model	MonuMAI (Y-Acc)	PASCAL (Y-Acc)
Ours (Joint-N)	95.71	89.25
Ours (Sequantial-N-Int)	92.77	80.61
Sequential [28]	92.85	82.8
Joint [28]	**97.69**	85.57
Greybox [3]	94.04	88.30
EXPLANet [6]	90.40	82.4
DeiT-B [3, 33]	96.48	**90.85**
MonuNet [21]	83.11	–

Best results for each evaluation measurement are in bold. MonuNet was designed and proposed specifically for monument style classification

dataset itself but also on the architecture of the model, to use first-order-logic to understand the model reasoning process and to intervene in it.

Additionally, we achieved state-of-the-art competitive results despite our model's lower complexity compared to most of the other transparent approaches as we did not make use of object detection or semantic segmentation. A classifier relying on an object detector like EXPLANet [6] necessitates complex architectures such as Faster R-CNN [14] or RetinaNet [23], further escalating the training complexity. Similarly, employing a segmentation model like DeepLab-V3+ [4] as done in Greybox [3] demands significant resources; in fact, note that a model based on DeepLab-V3 requires over 101 layers when employing ResNet-101 [16] as a backbone, whereas our model utilizes fewer than 60 layers. Furthermore, training an object detector or a segmentation model requires complex annotations, such as bounding boxes or semantic mask annotations, which must be drawn by experts.

4.9 The model as explainable AI model

With the aim of analysing and discussing the use of our proposed approach as XAI model, we refer to [26] who introduced some key considerations that should be made when implementing new explainable AI techniques and models. Below we resume the considerations and discuss how our proposed approach fulfils them.

- Contrastive explanations: explanations are more effective when presented in a contrastive manner. This involves explaining not only why decision X was made, but also why was decision X preferred over decision Y. The visualization of the making-decision process allows the user to understand not only which concepts contributed in a positive way to the decision, but also which other concepts contributed in a negative way. Moreover, through exploration of the decision tree, users can analyse what alterations would be required for the decision tree to make a different decision. That is why we affirm that the proposed model satisfies this first requirement.
- Probabilities: grounding explanations in causal relationships is more effective than relying on probabilities. The use of probabilities alone to justify the choice of decision X lacks effectiveness unless complemented with causal connections. The combinations of first-order logic rules and concepts are powerful causal links that are intuitive for the user and helpful to understand the decision made. Furthermore, these decisions can be visualized by the user as decision paths.
- "Explanations are social": the author remarks on the character of explanations as a transfer of knowledge as the result of an interaction. This interaction with the user is the result of the learning cycle proposed and implemented in this research work. The user is able to define background knowledge and constraints before the training starts. During the training phase the user is able to analyse and control the model. Based on that analysis and control, the learning process can be intervened by redefining the knowledge in form of new rules and constraints. The user can even modify the routing process, this is, the making-decision process as we have shown in Section 4.7. Additionally, our model is compatible with the user concept intervention as shown in [18].

5 Conclusion

In this research work, we explored the fusion of soft decision trees, neural symbolic learning, and concept learning, resulting in an interpretable classification model that bases its decisions on human-understandable concepts and enables user intervention and the incorporation of expert knowledge.

One of the key advantages of our approach is the ability to define constraints to the routing process of the soft decision tree. These constraints are specified in the form of first-order logic rules and predicates that are based on the combinations of concepts or classes. This empowers users to have greater control over the decision-making process (i.e. which nodes/leaves to visit for specific classes or in response to the presence of certain concepts). By incorporating this level of control, the model becomes highly adaptable and customizable to meet specific requirements and preferences. The definition of rules and predicates enables not only the user intervention in training time but also the posterior inspection of the model's reasoning.

All of this results in an interpretable concept-based architecture capable of incorporating expert knowledge and enabling user control and intervention. We test our proposed approach in two challenging datasets and compare it to state-of-the-art solutions, achieving competitive results and surpassing the state-of-the-art results for transparent models on the PASCAL dataset.

In future work, we will continue exploring the potential of combining neural symbolic learning and soft decision trees. Our approach could enhance the model's interpretability, transparency, and adaptability. This makes it a powerful tool for decision-making in the domain of image classification and others such as in the field of reinforcement learning, where the use of soft decision trees has been widely explored. Additionally, we believe that combining our work with other techniques such as pruning techniques could improve the transparency of our model and optimize it. Our proposed solution, as any other supervised concept learning model, requires concept annotations. Some authors have explored solutions such as the automatic extraction of concepts [13, 20, 30]. We believe that the combination of some of these methods with our proposed solution is an interesting future task.

Finally, we believe that further research must be conducted in order to improve the model-human interaction in the field of deep learning with the aim of increasing trust in AI models.

Acknowledgements Partial financial support was received from HAT.tec GmbH. The funders had no role in the study design, data collection, analysis, and preparation of the manuscript.

Author Contributions The authors confirm contribution to the paper as follows: study conception, design, analysis and interpretation of results: David M. Rodríguez; draft manuscript preparation: David M. Rodríguez, Manuel P. Cuéllar; Direction: Manuel P. Cuéllar, Diego P. Morales. All authors reviewed the results and approved the final version of the manuscript.

Funding Funding for open access publishing: Universidad de Granada/CBUA.

Data availability and access The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical and informed consent for data used Not applicable.

References

1. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 58:82–115
2. Samy B, Garcez ADA, Serafini L, Spranger M (2022) Logic tensor networks. Artif Intell 303:103649
3. Bennetot A, Franchi G, Del Ser J, Chatila R, Díaz-Rodríguez N (2022) Greybox xai: a neural-symbolic learning framework to produce interpretable predictions for image classification. Knowl-Based Syst 258:109947
4. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Computer vision–ECCV 2018: 15th European conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII 15, Springer, pp 833–851
5. Chen X, Mottaghi R, Liu X, Fidler S, Urtasun R, Yuille A (2014) Detect what you can: detecting and representing objects using holistic models and body parts. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1971–1978
6. Díaz-Rodríguez N, Lamas A, Sanchez J, Franchi G, Donadello I, Tabik S, Filliat D, Cruz P, Montes R, Herrera F (2022) Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: the monumai cultural heritage use case. Inf Fusion 79:58–83
7. Ding Z (2019) Popular-rl-algorithms. https://github.com/quantumiracle/Popular-RL-Algorithms
8. Donadello I, Serafini L (2016) Integration of numeric and symbolic information for semantic image interpretation. Intelligenza Artificiale 10(1):33–47
9. Donadello I, Serafini L, Garcez ADA (2017) Logic tensor networks for semantic image interpretation. In: Proceedings of the 26th international joint conference on artificial intelligence, pp 1596–1602
10. Došilović FK, Brčić M, Hlupić N (2018) Explainable artificial intelligence: a survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, pp 0210–0215
11. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The PASCAL Visual Object Classes Challenge (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html
12. Frosst N, Hinton G (2017) Distilling a neural network into a soft decision tree. arXiv:1711.09784
13. Ghorbani A, Wexler J, Zou JY, Kim B (2019) Towards automatic concept-based explanations. Adv Neural Inf Process Sys 32

14. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
15. Hazimeh H, Ponomareva N, Mol P, Tan Z, Mazumder R (2020) The tree ensemble layer: differentiability meets conditional computation. In: International conference on machine learning, PMLR, pp 4138–4148
16. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE Conference on computer vision and pattern recognition, pp 770–778
17. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. 3rd International conference on learning representations, pp 1–15
18. Koh PW, Nguyen T, Tang YS, Mussmann S, Pierson E, Kim B, Liang P (2020) Concept bottleneck models. In: International conference on machine learning, PMLR, pp 5338–5348
19. Kontschieder P, Fiterau M, Criminisi A, Bulo SR (2015) Deep neural decision forests. In: Proceedings of the IEEE international conference on computer vision, pp 1467–1475
20. Kumar A, Sehgal K, Garg P, Kamakshi V, Krishnan NC (2021) Mace: model agnostic concept extractor for explaining image classification networks. IEEE Trans Artif Intell 2(6):574–583
21. Lamas A, Tabik S, Cruz P, Montes R, Martínez-Sevilla Á, Cruz T, Herrera F (2021) Monumai: dataset, deep learning pipeline and citizen science based app for monumental heritage taxonomy and classification. Neurocomputing 420:266–280
22. Lamberti F, Morra L, Miro FD (2021) End-to-end training of logic tensor networks for object detection
23. Lin T-Y, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE International conference on computer vision (ICCV), Oct
24. Lockhart J, Magazzeni D, Veloso M (2022) Learn to explain yourself, when you can: equipping concept bottleneck models with the ability to abstain on their concept predictions. arXiv:2211.11690
25. Martone S, Manigrasso F, Lamberti F, Morra L (2022) Prototypical logic tensor networks (proto-ltn) for zero shot learning. In: 2022 26th International conference on pattern recognition (ICPR), IEEE, pp 4427–4433
26. Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell 267:1–38
27. Molnar C (2020) Interpretable machine learning. Lulu. com,
28. Morales Rodríguez D, Pegalajar Cuellar M, Morales DP (2023) On the fusion of soft-decision-trees and concept-based models. Available at SSRN 4402768
29. Mutahar G, Miller T (2022) Concept-based explanations using non-negative concept activation vectors and decision tree for cnn models. arXiv:2211.10807
30. Posada-Moreno AF, Surya N, Trimpe S (2023) Extracting concepts with local aggregated descriptors. Eclad
31. Ribeiro MT, Singh S, Guestrin C (2016) Why should I trust you? Explaining the predictions of any classifier. ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
32. Speith T (2022) A review of taxonomies of explainable artificial intelligence (xai) methods. In: 2022 ACM Conference on fairness, accountability, and transparency, pp 2239–2250
33. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jegou H (2021) Training data-efficient image transformers &amp; distillation through attention. In: Meila M, Zhang T (eds) Proceedings of the 38th international conference on machine learning, vol 139 of proceedings of machine learning research, pp 10347–10357. PMLR, 18–24. https://proceedings.mlr.press/v139/touvron21a.html
34. Wagner B, d'Avila Garcez AS (2021) Neural-symbolic integration for fairness in ai. In: CEUR Workshop Proceedings, vol 2846
35. Wagner B, Garcez ADA (2022) Neural-Symbolic Integration for Interactive Learning and Conceptual Grounding. http://arxiv.org/abs/2112.11805. arXiv:2112.11805
36. Wan A, Dunlap L, Ho D, Yin J, Lee S, Jin H, Petryk S, Bargal SA, Gonzalez JE (2020) Nbdt: neural-backed decision trees
37. Wang S, Fan Y, Jin S, Takyi-Aninakwa P, Fernandez C (2023) Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries. Reliab Eng Syst Saf 230:108920
38. Zarlenga ME, Barbiero P, Shams Z, Kazhdan D, Bhatt U, Jamnik M (2022) On the quality assurance of concept-based representations. https://openreview.net/forum?id=Ehhk6jyas6v
39. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. IEEE Conference on computer vision and pattern recognition, pp 2921–2929