Applied Soft Computing Journal 160 (2024) 111632

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

On the fusion of soft-decision-trees and concept-based models ☆

David M. Rodríguez [a,b,*], Manuel P. Cuéllar [a], Diego P. Morales [a]

[a] University of Granada, 18071 Granada, Spain
[b] HAT.tec Lilienthalstraße 15, 85579 Neubiberg, Germany

ARTICLE INFO

Keywords:
Soft decision trees
Concepts
XAI
Image classification

ABSTRACT

In the field of eXplainable Artificial Intelligence (XAI), the generation of interpretable models that are able to match the performance of state-of-the-art deep learning methods is one of the main challenges. In this work, we present a novel interpretable model for image classification that combines the power of deep convolutional networks and the transparency of decision trees. We explore different training techniques where convolutional networks and decision trees can be trained together using gradient-based optimization methods as usually done in deep learning environments. All of this results in a transparent model in which a soft decision tree makes the final classification based on human-understandable concepts that are extracted by a convolutional neural network. We tested the proposed solution on two challenge image classification datasets and compared them with the state-of-the-art approaches, achieving competitive results.

1. Introduction

Nowadays, the potential of convolutional deep learning models for the task of image classification has been proven. However, many of these models are considered *black-box* models as they can be opaque to the users due to the absence of any mechanism to explain the decision-making process [1], such as Artificial Neural Networks (ANNs). To achieve a higher degree of transparency and interpretability, new techniques and models have been proposed in recent years with the aim of developing more interpretable artificial intelligence [2]. Most of the solutions and models proposed in recent years can be classified into two categories: transparent models and post-hoc explainability techniques [2,3]. Post-hoc explainability techniques are popular methods in the field of deep learning. Some of the most known techniques belonging to this category are LIME [4], which perturbs the input and demonstrates how the predictions change, or Grad-CAM [5], which is used in neural networks and uses the gradients to produce a map, highlighting the important regions in the image for predicting the class. On the other hand, the creation of transparent models is one of the main goals of XAI, but it is still a distant goal in the field of deep learning.

Classical decision trees are among the best-known machine learning algorithms and have been widely used to solve machine learning tasks such as classification or regression problems. Moreover, they are considered transparent and interpretable machine learning models, as users can visualize and trace the decision-making process or extract if-then rules that explain the decision process [6]. However, integrating classical decision trees with deep learning methods is not straightforward as they are not differentiable. The employment of soft decision trees is getting growing interest as a potential solution [6–9]. Soft decision trees are models inspired by classical decision trees and that conserve the structure formed by nodes, edges, and leaves. The key difference relies on the fact that they perform probabilistic routing (or soft routing) instead of deterministic routing, which makes them differentiable.

In this research article, we explore the use of decision trees in a deep learning environment. The goal of this article is to present a novel image classification method where the power of convolutional neural networks and the transparency of decision trees are combined, resulting in an interpretable model in which image classification is based on human-understandable concepts. Our proposed solution is a concept-based model, developed as a fusion of soft decision trees and a deep convolutional neural network. It is based on concept bottleneck models and can be trained with classical gradient-based optimization techniques as known from deep learning. The decision-making process is transparent to the user and makes our models interpretable. Furthermore, we test the proposed approach on two challenging datasets and achieve competitive results compared to the state-of-the-art.

The contributions of this research work are as follows:

1. We provide a comprehensive overview of the current state of research in this area.

☆ 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.

* Corresponding author at: University of Granada, 18071 Granada, Spain.
E-mail addresses: dmorales@correo.ugr.es, david.morales@hattec.de (D.M. Rodríguez), manupc@ugr.es (M.P. Cuéllar), diegopm@ugr.es (D.P. Morales).

https://doi.org/10.1016/j.asoc.2024.111632
Received 7 March 2023; Received in revised form 20 February 2024; Accepted 9 April 2024
Available online 3 May 2024

2. We explore the combination of decision-trees with deep learning models.
3. We proposed a new interpretable model, that results from the fusion of a concept extractor and a soft-decision tree.
4. We analyze and compare different training approaches for the proposed solution.
5. We explore related works and compare the proposed solution to the state-of-the-art methods.

The manuscript is organized as follows. Section 2 includes an overview of the state of the art. Section 3 presents the proposed methods and training approaches. Section 4 introduces the two datasets, describes the experiments carried out, and analyzes the obtained results. Finally, Section 5 closes with conclusions and future lines of research.

2. Related work

While the very first machine learning algorithms were easily interpretable, in the last few years deep neural networks (DNNs) have become the standard solution to many tasks [2,10]. DNNs are the state of the art in many machine learning problems because of their great generalization. However, they are considered *black-box* machine learning models as the decision-making process is opaque to the user, who cannot get an explanation of the decisions made by the model [1]. In this context, there has been a growing interest on explainable artificial intelligence (XAI). Post-hoc explanations, which refer to the use of interpretation methods after training a model and feature relevance methods are the most adopted approaches to explain DNNs [2]. Most of these explanation techniques provide heat maps or saliency maps to identify the regions of the input images that networks look at when making predictions. Some well-known visual explanation techniques are Class Activation Mapping (CAM) [11] or Local Interpretable Model-Agnostic Explanations (LIME) [4].

On the other hand, the definition of transparent deep learning models is one of the main goals of XAI and an active research field. The original problem lies in the fact that historically there has been a trade-off between power and interpretability or transparency of the proposed models [12]. Classical machine learning models and algorithms, such as decision trees or k-NN, are interpretable and transparent, but they are outperformed by opaque models, such as deep neural networks. That is why recent research has focused on addressing this well-known performance-explainability trade-off [13,14] and defining models that are transparent by design and that do not need post-hoc explanations techniques.

Decision Trees are a classical machine learning algorithm based on if-then-rules. These decision rules are presented in a branch-based graph that is followed in order of making the final prediction. These models are considered as transparent models as following those paths or rules enables humans to understand why a prediction or a classification is made [15]. However, as already mentioned, decision trees do not generalize as well as neural networks. Some research has been done to explore how decision trees can be improved and adapted to be used to solve deep learning problems. Kontschieder et al. [16] presented Deep Neural Decision Forests where they aimed to combine representation learning as known from deep architectures with the divide-and-conquer principle of decision trees. They introduced a stochastic and differentiable decision tree -neural decision tree- and constructed their proposed solution as an ensemble of those neural decision trees. In other words, a decision forest provides the final predictions. Wan et al. [17] presented a hierarchy-learning-based model called Neural-Backed Decision Trees where they proposed to replace the network's final linear layer with a decision tree, inducing hierarchies that shall be used to explain the decision of the model. Frosst and Hinton [7] proposed distilling a neural network into a Soft-Decision-Tree. The authors described a method for using a trained neural net to train a soft decision tree by stochastic gradient descent using the predictions of the neural net as targets. Given an input, their model makes hierarchical decisions based of the learned filters and selects as output a particular static probability distribution over classes.

In the search for more transparent models, another approach that has been studied is concept-based explainability. The authors who explore this approach aim to develop interpretable models designing them to base their decisions on concepts, where concepts are considered high-level and semantically meaningful units of information commonly used by humans to explain their decisions. This approach enables us to interpret the reasoning process by generating explanations based on those concepts [18]. Furthermore, this approach can allow users to improve the performance of a model through concept interventions, in which mispredicted concepts are corrected using expert knowledge [19,20]. A well-known article in this field was presented by Alvarez Melis and Jaakkola [21], who proposed the self-explaining neural networks (SENN). Their model consists of a concept encoder, a relevance score generator, and an aggregation function. They proposed to define concepts using an autoencoder and trained their model to use these concepts for classification. The decisions of the model can be explained by looking at the scored concepts, without the need of post-hoc explanation techniques. However, the challenge of this approach is finding understandable and appropriate concepts. The concepts could be defined by an expert, but this would require data annotations and human intervention. The use of human-provided concepts has been studied. Some studies based on this approach trained supervised models with annotated concepts predefined by human specialists such as the colors and shapes of objects, which are precise and accurate for human understanding [10]. Koh et al. [19] proposed concept bottleneck models (CBMs). In these models, the classification task is performed in two steps. A first CNN model works as a concept extractor and maps raw inputs (x) to concepts (c), and a second model performs the final classification by mapping these concepts (c) to targets (y). Some authors propose to train object detection or segmentation models to localize object parts and combine those models with a classifier to build an interpretable model that bases its decision on the detected object parts [10,22].

In this article, we investigate the use of decision trees in combination with deep learning methods. We believe that concept-based learning is one of the most promising approaches in the field of interpretable deep learning. However, we identify a lack of transparency in how the decision-making process once the concepts are defined. For that reason, we explore how to combine and train decision trees and concept-based models and 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. This last point opens the door to human intervention, as an expert could explore the decision tree and improve it by using his knowledge to redefine the decision tree.

3. Methodology

In this article, we study the fusion of Soft Decision Trees and Concept Bottlenecks. We propose to use a CNN as a concept extractor to map the image to concepts as proposed in [19], following the approach presented in Fig. 1. A Soft Decision Tree is used as a predictor, which performs the final classification based on the extracted concepts.

3.1. Concept bottlenecks

The classification problem is divided into two subtasks: concept extraction and classification (see Fig. 1). The concept extraction task is defined as a multilabel classification problem. The labels (concepts) depend on the dataset, which should be annotated accordingly. After

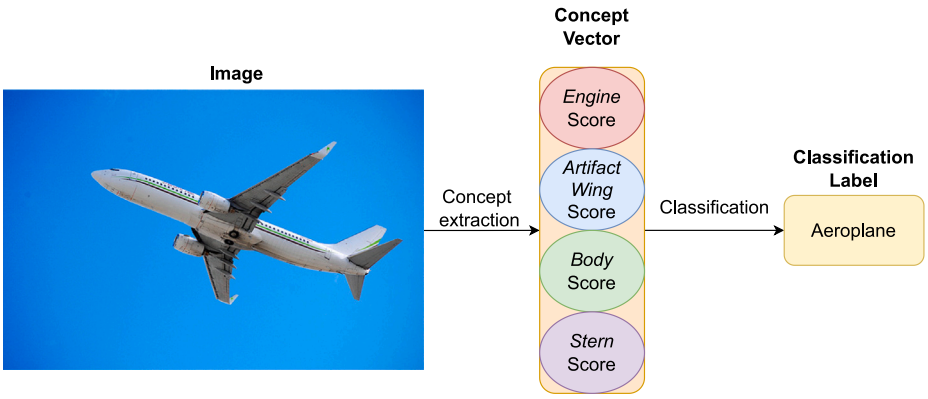

Fig. 1. Diagram showing a classification task resolved using concept learning. The task is divided into two subtasts: first, a concept extraction takes place producing a concept vector as output. This extraction can be implemented as a multilabel classification problem, where the labels depend on the datasets. The concept vector contains the scores for each label. Second, a classification process takes place based on the concept vector.

the extraction, the classification takes place based on the concept vector obtained.

To formalize the proposed solution, we define the classification problem as follows: Consider an input $x \in \mathbb{R}^d$, a target output $y \in \mathbb{Y}$ and a vector of concepts $\mathbf{c} \in [0,1]^k$, such that the training samples compose a set of the form $[(x_n, y_n, \mathbf{c}_n); n = 1...N]$. The proposed model is of the form $t(g(x))$, where $g : \mathbb{R}^d \to [0,1]^k$ maps from input space to concept space and $t : [0,1]^k \to \mathbb{Y}$ is a decision tree that maps from concept space to target space. To train the model, two loss functions are defined: a first loss function $L_Y : \mathbb{Y} \times \mathbb{Y} :\to \mathbb{R}_+$ that given a training sample $(x_i, \mathbf{c}_i, y_i)$ measures the discrepancy between the output of the model $y' = t(g(x_i))$ and the target output y_i. This is a multi-class classification task so we use the multi-class cross-entropy loss as it is the standard solution for these tasks. The second loss function is of the form $L_c : [0,1]^k \times [0,1]^k \to \mathbb{R}^+$ measures the discrepancy between the output of the concept extractor $g(x_i)$ and the true vector of concepts $\mathbf{c}_i$. This is the multi-label classification task so we use the binary cross-entropy loss in this case.

3.2. Soft decision trees

In classical decision trees, every sample is routed to exactly one direction at every node (deterministic routing or hard routing), which introduces discontinuities in the loss function and makes classical decision trees not continuously optimizable [23]. For this reason, classical decision trees cannot be trained using gradient descent-based algorithms. That is the reason why we decided to explore the use of binary soft decision trees, more specifically, our model is based on the model proposed in [7]. These soft decision trees can be trained with mini-batch gradient descent as they perform probabilistic routing (or soft routing) instead of deterministic routing, avoiding the introduction of discontinuities in the loss function and making them continuously optimizable [7,23].

Soft decision trees are composed of nodes and leaves, just as classical trees. Each inner node i has a learned filter w_i and a bias b_i. Given an input feature x the probability of passing to the right branch at the inner node i is:

$$p_i(x) = \sigma(xw_i + b_i) \tag{1}$$

where σ is the logistic sigmoid function. Since this model is a binary tree, $1 - p_i(x)$ is the probability of routing to the left branch. In Fig. 2 we illustrate the structure of an inner node of a binary soft decision tree and the routing process that would take place in the inner node i for an input x. In the figure on the right we assume some values for the input and for the weights and biases and calculate the output of the routing process.

The probability $P^l(x)$ of arriving at leaf node l given the input x is

$$P^l(x) = \prod_N p_i(x)^{\mathbf{1}[l \swarrow i]} (1 - p_i(x))^{\mathbf{1}[i \searrow l]} \tag{2}$$

The notation $\mathbf{1}$ represents an indicator function that produces one if the condition holds and zero otherwise. The notation $[l \swarrow i]$ (and $[i \searrow l]$) indicates leaf l belongs to the left (resp. right) subtree of node i. Each leaf node l produces a probability distribution over the possible output classes

$$Q^l = softmax(\phi^l) \tag{3}$$

where ϕ^l is a learned parameter. The output of the model is the distribution at the leaf with the maximum path probability. In Fig. 3 we illustrate all these equations by providing an example of how to calculate the probabilities and outputs. We show a soft decision tree with one hidden layer for a binary classification problem. We assume some values for the input $x = 1$ and for the weights w_i and biases b_i and demonstrate the decision process that would take place. We assume that for the second leaf the learned distribution is $Q^1 = [0,2; 0.8]$, so for that leaf, the second class would be selected. We compute the probabilities $p_i = p_i(x) = \sigma(xw_i + b_i)$ as shown in Fig. 2. Computing the probabilities $P^l(x)$ of arriving at each of the leaves, it can be seen that the highest probability is given for the first leaf: $max_i P^i(x) = P^1(x) = \prod_2 p_i(x)^{\mathbf{1}[l \swarrow 1]} (1 - p_i(x))^{\mathbf{1}[1 \searrow l]} = (0.75^1 * 0.25^0) * (0.21^0 * 0.79^1) = 0.59$. This implies that for the input x, the output of the tree is $Q^1 = softmax(\phi^1)$, where ϕ^1 is a learned probability distribution over the output classes (two in our case) for the given leaf.

The decision tree is trained using the loss function

$$L_T(x) = -\sum_{l \in L} P^l(x) \sum_{k \in Y} y_k \log Q^l_k \tag{4}$$

where Y is the set of possible labels, k is the index of the label, and y_k is the observed probability of x being categorized as k, which is either 0 or 1. Observe that this is just the classical cross-entropy function for each leaf, weighted by its path probability. Using again the example in Fig. 3, we illustrate how the loss would be calculated: we assumed the learned distribution $Q^1 = [0,2; 0.8]$, so for that leaf, the second class would be selected and we assume now that the classification is correct (x belongs to the second class), the partial loss for leaf 1 would be $L_{l_1}(x) = P^1(x) \sum_{k \in Y} y_k \log Q^1_k = 0.59\ (0 \log 0.2 + 1 \log 0.8) = -0.057$. To calculate the total loss $L_T = -\sum_{l \in L} L_l$ we would have to do the same calculations for each leaf and sum them, as shown in Eq. (4).

3.3. Overall structure

In Fig. 4 we show the overall structure of the proposed model.

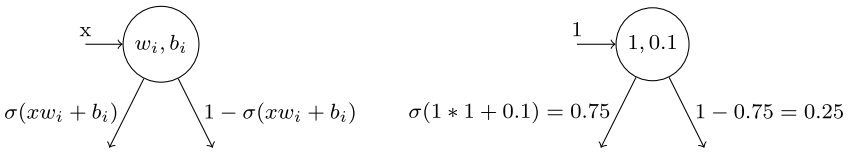

Fig. 2. On the left side we present an inner node i of a binary soft decision tree. Each inner node has two learned parameters associated: the weight w_i and a bias b_i. On the right side we illustrate the routing process according to Eq. (1) with an example. To this aim the variables are given the following values: $x = 1, w_i = 1, b_i = 0.1$.

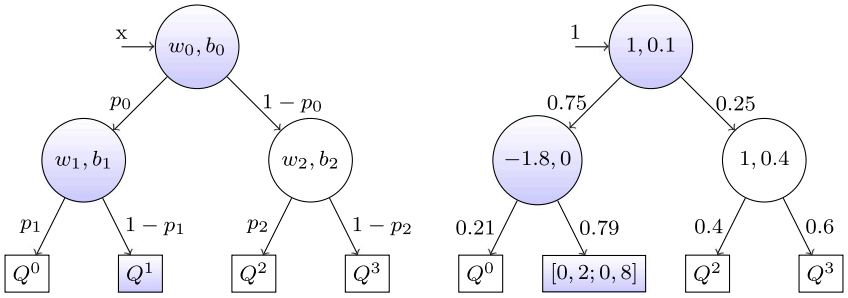

Fig. 3. On the left figure we show a soft decision tree with one hidden layer for a binary classification problem. In the figure on the right, we assume some values for the input x and for the weights w_i, biases b_i, and learning variables like $Q^1 = [0, 2; 0.8]$. The probabilities of routing to the left or to the right are shown for each level.

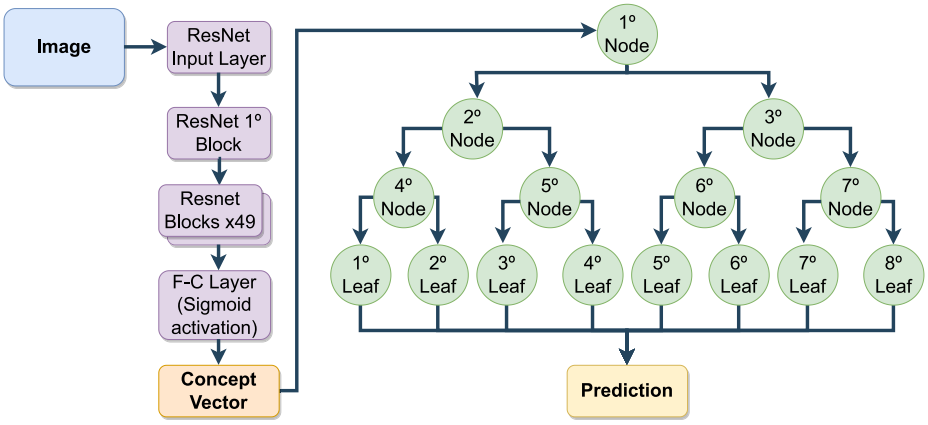

Fig. 4. Overall structure of the proposed model. The concept extractor g is implemented by a Resnet-50. Its final layer is implemented by a fully connected layer with a sigmoid activation function. The concept extractor gets an image as input and outputs the concept vector. The binary soft decision tree t takes the concept vector as input and outputs the final prediction. We draw a tree with just four levels since adding more levels would result in an excessively large figure. FC is the abbreviation for fully connected.

3.4. Training environment

In this section, we describe different methods for training the proposed method. We analyze and study the three different ways of training a concept bottleneck model that were proposed by Koh et al. [19]:

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

In our case, L_c is the concept loss function described in Section 3.1 while L_Y correspond to the loss function described in Eq. (4). Compared to the independent model the idea of the sequential model is to allow the final classifier t to adapt itself to a given extractor. On the other hand, the idea of the joint model is to allow the refinement of the concept extractor in order of improving the performance of the main task.

4. Experimental setup, evaluation and results

In this section, we present two datasets used to evaluate the proposed method. Next, we describe the implementation details as well as the experiments carried out, including the evaluation metrics considered. Finally, we report and analyze the results obtained.

4.1. Datasets

We evaluated the proposed methods on the MonuMAI dataset [24] and on the Semantic PASCAL-Part dataset [25].

The MonuMAI dataset [24] is an image dataset that contains more than 1500 images of monuments belonging to four architectural styles: Gothic, Hispanic-Muslim, Renaissance and Baroque. This dataset was labeled by human experts who generated annotations for monument style classification and key architectural element detection. The experts also generated labels for fifteen key architectural element types (i.e. lobed arch, trefoil arch, solomonic column...). The classification

and analysis of those key elements can be seen as a necessary subtask when classifying monuments by their architectonic style and should be an argument when explaining the decision of a classifier as done in [22,24].

The PASCAL VOC 2010 dataset [26] is a well-known image dataset organized into 20 object classes. The PASCAL-Part dataset [27] provided additional annotations for the PASCAL VOC 2010 dataset. In this research work, we use a curated version of the PASCAL-Part dataset provided by Díaz-Rodríguez et al. [10] and based on the Semantic PASCAL VOC dataset [25]. In this version of the dataset, the number of object part categories is reduced by aggrouping some similar categories into a main one (i.e "right leg" and "left leg" could be reduced into a single category "leg"). Furthermore, the authors selected the images so that only one main object class per image was present(classical image classification problem). This dataset contains more than 1400 color images including 20 categories (i.e Person, TV, Train, etc.) and more than 40 different parts (i.e Leg, Body, Wheel, ...), where each image has only one associated category. This dataset has also been used to test concept-based or part-based models [10,22].

4.2. Implementation details

The proposed method was implemented on Pytorch, and the code is available for download.[1] We use a Resnet-50 [28] as the backbone for the concept extractor for all methods. Its final layer is implemented by a fully connected layer with a sigmoid activation function. To implement the three different ways of training a concept bottleneck, we adapt the code provided by the authors of the original article [19]. In order to make a first comparison with baseline methods, we trained the three different concept bottleneck approaches based on the prior models using a multilayer perceptron with one hidden layer for the classification net. We kept the Resnet-50 as concept extractor for the baseline 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. Our soft decision tree is based on the implementation provided in[2] for the model described in [7]. After a preliminary analysis, we decided to set the depth parameter of the soft decision trees to 5. We used the Adam optimization algorithm [29] for all networks.

4.3. Experiments

This section describes the experiments designed to evaluate the proposed methods (see Section 3). We individually tested the proposed approach on both datasets and compared them to the baseline methods. In order to compare our results with the state-of-the-art models, we kept the splits in training and test sets that were proposed in [10,22] for the two considered datasets (see Section 4.1). Then, we trained the model using the training set. To evaluate the performance of the proposed models and make a fair comparison with other approaches, we evaluated the proposed solution on the isolated test and computed some popular metrics for classification.

4.4. Results

In this section, we report and analyze the results obtained in the experiments described in Section 4.3. In this section we present the results obtained for the different methods on the proposed datasets and compare them to the baseline methods. As proposed in [19], we evaluate how each proposed approach performs for two different tasks: concept extraction and final classification (the main task), using the metrics proposed by the authors. 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

$$\text{Y-Acc} = Acc(y, y') \tag{5}$$

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

$$\text{C-Error} = 1 - \frac{avg(BinaryAcc(c_i, c_i'))_{i \in 1..N}}{100} \tag{6}$$

where c_i and c_i' are the components of the vectors c, the vector representing the annotated concepts for a given sample x, and $c' = g(x)$ the vector representing the prediction of g for the sample x. We repeated every experiment 30 times and present the mean results with standard deviation in Table 1 for the MonuMAI dataset and in Table 2 for the PASCAL dataset.

Table 1
Results of the proposed experiments on the MonuMAI dataset. Best results for the final classification task are in bold.

MonuMAI			
	Ind-Tree	Seq-Tree	Joint-Tree
Y-Acc	92.38 ± 0.41	92.74 ± 0.21	$\mathbf{97.82 \pm 0.46}$
C-Error	0.025 ± 0.001	0.025 ± 0.001	0.029 ± 0.002
	Ind-Baseline	Seq-Baseline	Joint-Baseline
Y-Acc	92.49 ± 0.47	92.51 ± 0.22	95.51 ± 0.55
C-Error	0.025 ± 0.001	0.025 ± 0.001	0.076 ± 0.004

Table 2
Results of the proposed experiments on the PASCAL dataset. Best results for the final classification task are in bold.

PASCAL			
	Ind-Tree	Seq-Tree	Joint-Tree
Y-Acc	84.29 ± 0.39	84.36 ± 0.34	$\mathbf{85.14 \pm 0.53}$
C-Error	0.028 ± 0.001	0.028 ± 0.001	0.029 ± 0.001
	Ind-Baseline	Seq-Baseline	Joint-Baseline
Y-Acc	84.26 ± 0.5	84.26 ± 0.39	83.06 ± 0.92
C-Error	0.028 ± 0.001	0.028 ± 0.001	0.035 ± 0.01

It can be observed that the Independent (Ind) models and the Sequential (Seq) models performed very similarly on both datasets. Please note that the C-Error for those two approaches is the same for the proposed approach and for the baseline models as the same concept extractor g is used for both models and only t is different. Please see 3. For the MonuMAI dataset, the Joint-Tree model gets the best results on the main task, achieving over 2 points accuracy more than the second-best model. The sequential tree model performs slightly better than the corresponding baseline model while the independent models perform very similar. Performed t-tests showed that the improvement is statistically significant for the Joint-Tree models with respect to the baseline model and to the second best model (Seq-Tree). All tests were performed for a significance level $\alpha = 0.05$. Regarding the concept prediction tasks, the Joint-Tree model and the concept extractor trained for the independent and the sequential models get similar results and outperformed the Joint-Baseline model. Regarding the PASCAL dataset, the best results for the main task are obtained for the approach Joint-Tree. The improvements with respect to the Joint-Baseline model and with respect to the second best model (Seq-Tree) are statistically significant. All tests were performed for a significance level $\alpha = 0.05$. Regarding the secondary task, the Joint-Tree model performs very similar to the concept extract trained for the Independent and for the Sequential approaches, outperforming the Joint-Baseline model also for this task. On resume, on the main task the Joint-Tree model performs statistically significantly better than any of the other

[1] https://github.com/DavidMrd/SoftConceptTree
[2] github-decisiontree.

Table 3
Results compared to the state of the art. Best results for each evaluation measurement are in bold. MonuNet was designed and proposed specifically for monument style classification.

Model	MonuMAI (Y-Acc)	PASCAL (Y-Acc)
Independent (Ours)	92.38	84.29
Sequential (Ours)	92.74	84.36
Joint (Ours)	**97.82**	85.14
Greybox [22]	94.04	88.30
EXPLANet [10]	90.40	82.4
DeiT-B [22,30]	96.48	**90.85**
MonuNet [24]	83.11	–

models on both datasets. For the secondary task, the Joint-Tree model outperforms the Joint-Baseline model and gets very similar results to the concept extractors although these seconds were trained exclusively for that task. For all these reasons we choose this model as our proposed approach over the other models.

All performed t-tests that were referenced in this section can be found in Appendix.

4.5. Compare to the state-of-the-art

In this section, we compare our models and results to the state of the art. We compare the proposed models with four recently proposed approaches that were presented by different authors and introduced above in Section 2. Two of the models are transparent models (Greybox [22] and EXPLANet [10]) and the other two models are opaque models (DeiT-B [22,30] and MonuNet [24]). 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 3. On the MonuMai dataset, the Joint approach achieves higher accuracy than the second-best approach (DeiT-B [22,30]). On the PASCAL dataset, we achieve competitive results, and the Joint model is under the transparent approaches the one with the highest accuracy, performing slightly worse than the best model (DeiT-B). The Indepent and the Sequential models get competitive results on both datasets, performing better than MonuNet and EXPLANet. Compared to the transparent models, we achieved state-of-the-art competitive results although the complexity of our model is lower as we do not use object detection or semantic segmentation. Note that training an object detector or a segmentation model requires complex annotations such as bounding boxes or semantic mask annotations that experts should draw. Furthermore, a classifier based on an object detector such as EXPLANet [10] requires a complex architecture such as Faster R-CNN [31] or RetinaNet [32], which also increases the complexity of the training. The same issue occurs when a segmentation model such as DeepLab-V3+ [33] is needed, as for Greybox [22]. In fact, note that a model based on DeepLab-V3 has necessary more than 101 layers, as ResNet-101 [28] is used as backbone, while our model has less than 60 layers.

4.6. Discussion

4.6.1. Visualization

In Fig. 5 we visualize the decision-making process of the proposed soft-decision tree for a given image x. For the nodes that are visited during the inference process, we visualize the filter as a vector of 15 elements, where every element corresponds to one of the 15 concepts.[3] The symbol "−" represents that the presence of that concept would decrease the probability of taking the left path (increasing the probability of taking the right path), while the symbol "+" represents that the presence of that concept would increase the probability of taking the left path. Note that for a better understanding, we use a gray scale where the dark colors represent negative values and the light colors represent positive values. Given an image x, the concept extractor outputs the concept vector with which the soft decision tree is fed. In the case of the given sample x, the concept extractor has detected two concepts: broken pediment (position 11) and lintelled doorway (position 15). We can observe that for the first node, the presence of those elements increases the probability of taking the left path. The green arrows guide us throw the decision path to the first leaf node, which corresponds to class 3: Baroque.

In this way, the decision-making process is transparent to the user. Furthermore, a user or an expert could inspect the model and even would be able to edit the filter associated with any node. In this way, he could fix or improve the model by changing the weight of any concept on the decision of taking the left or the right path in a certain node. Also, the class associated to any leaf node could be modified if the expert considers that it is necessary. Furthermore, the decision of the concept extractor could be analyzed by using post-hoc explanation methods such as Grad-CAM [5] or LIME [4]. In Fig. 5, we demonstrate this option generating a saliency map for the concept "Broken pediment" which is present in the concept vector.

4.6.2. The model as explainable AI model

In order to discuss our proposed approach as explainable AI model, we refer to Miller [35] who introduced some considerations that should be taken into account when creating an explainable AI Mode.

- Contrastive explanations: explanations are more effective when presented in a contrastive manner. This involves explaining not only why making decision X, but also why choosing decision X instead of decision Y. We believe that our model fulfills this requirement as 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. Furthermore, by exploring the decision tree, the user can explore what should be different for the decision tree to make a different decision.
- Probabilities: relying on probabilities in explanations is less effective than referring to causes. Using probabilities to explain why choosing decision X is unsatisfying unless accompanied by causal links. We believe that the decision paths and the concepts are powerful causal links that are intuitive for the user and helpful to understand the made decision.
- "Explanations are social": the author remarks on the character of explanations as a transfer of knowledge as the result of an interaction. We believe that no interaction is possible with a model if it is not interpretable and transparent to the user. Decision trees are easy to understand through visualizations. This fact opens the door to interact with the model and to understand what would be the decision of it in different situations and in the presence or absence of different concepts. Additionally, our model is compatible with the user concept intervention as shown in [19]. Furthermore, modifying the weights of a given tree node allows the user to change the routing process, this is, the making-decision process. In other models where the user is not able to understand the decision-making process or the role of the different parameters and weights (i.e. in a neural network), this interaction is not possible.

The author added a fourth consideration about how humans rarely expect explanations to cover all causes of an event. In the field of concept learning, we believe that consideration should be addressed when selecting and annotating the concepts for a dataset.

[3] The concept vector represents the following elements: pointed arch, ogee arch, horseshoe arch, lobed arch, round arch, trefoil arch, solomonic column, flat arch, triangular pediment, segmental pediment, broken pediment, porthole, gothic pinnacle, serliana, lintelled doorway.

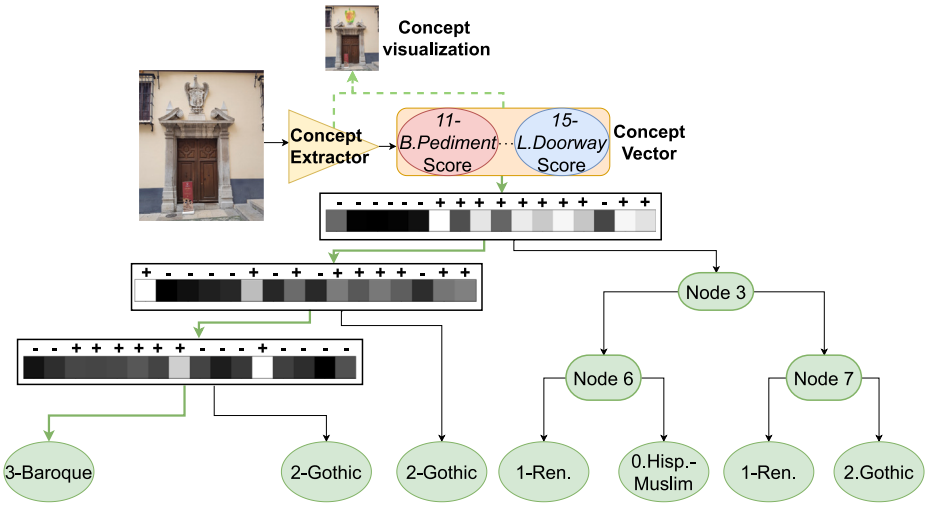

Fig. 5. Visualization of the prediction process. The green arrows indicate the path through the decision tree during the prediction process for the input x. Sample x is an image taken from the MonuMAI dataset [24]. We visualize the filters only for the nodes that are involved in the decision-making process for the given sample x. By following the decision path, we can observe that it leads to the first leaf node, which corresponds to class 3: Baroque. In the image, a Baroque lintelled doorway (position 15th in the concept vector) can be observed. This Baroque doorway was designed by Luis de Arévalo in the 18th century for the school "Colegio de San Fernando". The broken pediment (position 11th in concept vector) contains the shield of the Catholic Monarchs of Spain. In this example, we show how the output of the concept extractor can also be analyzed by using post-hoc techniques such as Grad-CAM [5]. In this example, a saliency map is generated for the concept "Broken pediment". Today, this doorway can be visited at the "Capilla Real" in Granada, Spain. [34].

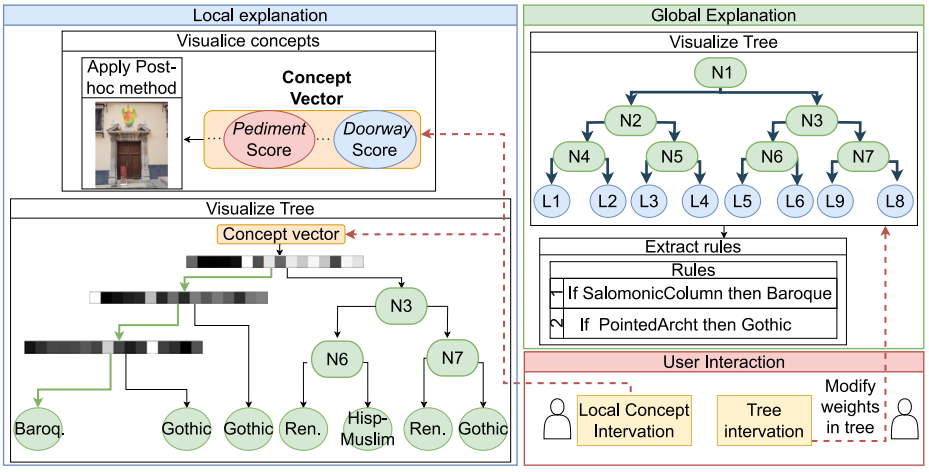

Fig. 6. User interface prototype: the user interface should offer functionality for three main tasks: global explanations, local explanations and user intervention.

4.7. User interface

In this subsection we discuss how a user interface should look like in order to implement and integrate all the ideas and methods introduced in this article so that a final user could benefit from our approach. In Fig. 6 we present a prototype of a user interface inspired by [36]. As it can be observed in the figure, we believe that the user interface should offer functionality for three main tasks: global explanations, local explanations and user intervention. For local explanations, given an input, the concept vector could be presented to the user who could understand in which concepts the decision was based. Furthermore, post-hoc explanation techniques could be applied to understand what the relevant features for each concept are, as was already explained before. The decision path could be presented to the user together with the filters, what would allow him to understand how the making-decision process was. In the context of global explanations, the user would be able to visualize and inspect the tree and observe the rules that could be extracted. Furthermore, the interface should also allow the intervention of the model at least in two ways: concept intervention as presented in [19] and tree intervention, where the user by visualizing the tree could update the weights to modify the decision paths.

5. Conclusion

In this research work, we explore the fusion of decision trees and deep learning models. We define an interpretable classification model using a decision tree that is able to perform classification basing its decision on human-understandable concepts. This is achieved by defining an architecture based on Concept Bottlenecks and Soft-Decision-Trees. The use of soft-decision trees allows us to train the models by using gradient-based optimization methods, as done when training classical deep-learning models. We explore different ways of training the model in a multitasking environment, forcing the model to use human-labeled concepts to perform the final classification. This all results in an interpretable concept-based architecture where the decisions are transparent to the user. We compare the proposed solution to the state-of-the-art

methods and achieve competitive results without the need of object detectors or object-part annotations.

In future work, we will continue exploring the potential of combining transparent models with deep learning models. We believe that other concept-based models and symbolic learning methods could profit from the lessons learned during this research work.

Our model, as most of the models based on concept learning, requires prior annotation of concepts. Although the datasets used in this article have required expert annotation for the use of concept learning, some authors have explored the automatic extraction of concepts [37–39]. We believe that the combination of some of these methods with our proposed solution in order for concepts to be extracted automatically is an interesting future task.

The use of soft-decision trees could make them more interpretable and self-explanatory, and the exploration of different training approaches could serve as inspiration for combining other interpretable and opaque approaches to explore more transparent architectures and models. Furthermore, we believe that by combining our work with other techniques such as pruning techniques or rule extraction techniques we could improve the transparency of our model and optimize it. Furthermore, our model opens the door to human intervention, where an expert is able to explore the model and even improve the decision-making process by modifying the decision tree. We believe that further research must be conducted in that direction in order to improve the user experience and the model-user interaction.

CRediT authorship contribution statement

David M. Rodríguez: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Conceptualization. **Manuel P. Cuéllar:** Writing – review & editing, Supervision, Project administration. **Diego P. Morales:** Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

All authors approved the version of the manuscript to be published.

Appendix. Statistical tests

See Tables 4–8.

Table 4
Paired t-test Joint-Tree compared to Joint-Baseline (MonuMAI).

Measure	Variable 1	Variable 2
Mean	97.821788	95.511552
Variance	0.207327031051035	0.29746714623724
Observations	30	30
Pearson correlation	0.108890922265716	
Observed mean difference	2.310236	
Variance of differences	0.450710135507586	
Degrees of freedom	29	
t Statistic	18.8481318943247	
$P(T \leq t)$	8.14210513605171E−18	
t Critical	2.0452296421327	

Table 5
Paired t-test Joint-Tree compared to Sequential-Tree (MonuMAI).

Measure	Variable 1	Variable 2
Mean	97.821788	92.739164
Variance	0.207327031051035	0.0450716466455179
Observations	30	30
Pearson correlation	−0.000136329516699961	
Observed mean difference	5.082624	
Variance of differences	0.252425034914484	
Degrees of freedom	29	
t Statistic	55.4092660719562	
$P(T \leq t)$	5.63100270443672E−31	
t Critical	2.0452296421327	

Table 6
Paired t-test Joint-Tree compared to Joint-Baseline (PASCAL).

Measure	Variable 1	Variable 2
Mean	85.1391316666667	83.0565193333333
Variance	0.277062102855746	0.854492516192644
Observations	30	30
Pearson correlation	−0.197720047131229	
Observed mean difference	2.08261233333334	
Variance of differences	1.32396273886678	
Degrees of freedom	29	
t Statistic	9.91359521249522	
$P(T \leq t)$	8.03302690253164E−11	
t Critical	2.0452296421327	

Table 7
Paired t-test Joint-Tree compared to Ind-Tree (PASCAL).

Measure	Variable 1	Variable 2
Mean	85.1391316666667	84.290657417301
Variance	0.277062102855746	0.153585186067178
Observations	30	30
Pearson correlation	−0.338992771103512	
Observed mean difference	0.848474249365627	
Variance of differences	0.570504112377189	
Degrees of freedom	29	
t Statistic	6.15275899509722	
$P(T \leq t)$	1.0483797811279E−06	
t Critical	2.0452296421327	

Table 8
Paired t-test Joint-Tree compared to Sequential-Tree (PASCAL).

Measure	Variable 1	Variable 2
Mean	85.1391316666667	84.3598615916955
Variance	0.277062102855746	0.118904640542591
Observations	30	30
Pearson correlation	0.104730424055795	
Observed mean difference	0.779270074971167	
Variance of differences	0.357948607137761	
Degrees of freedom	29	
t Statistic	7.13408513998445	
$P(T \leq t)$	7.50717148868144E−08	
t Critical	2.0452296421327	

References

[1] Warren J. von Eschenbach, Transparency and the black box problem: Why we do not trust AI, Philos Technol 34 (4) (2021) 1607–1622.

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[3] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, Dino Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. (CSUR) 51 (5) (2018) 1–42.

[4] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, Why should I trust you? Explaining the predictions of any classifier, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[5] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[6] Zihan Ding, Pablo Hernandez-Leal, Gavin Weiguang Ding, Changjian Li, Ruitong Huang, Cdt: Cascading decision trees for explainable reinforcement learning, 2020, arXiv preprint arXiv:2011.07553.

[7] Nicholas Frosst, Geoffrey Hinton, Distilling a neural network into a soft decision tree, 2017, arXiv preprint arXiv:1711.09784.

[8] Alizée Pace, Alex J. Chan, Mihaela van der Schaar, Poetree: Interpretable policy learning with adaptive decision trees, 2022, arXiv preprint arXiv:2203.08057.

[9] Pradyumna Tambwekar, Andrew Silva, Nakul Gopalan, Matthew Gombolay, Natural language specification of reinforcement learning policies through differentiable decision trees, IEEE Robot. Autom. Lett. (2023).

[10] Natalia Díaz-Rodríguez, Alberto Lamas, Jules Sanchez, Gianni Franchi, Ivan Donadello, Siham Tabik, David Filliat, Policarpo Cruz, Rosana Montes, Francisco Herrera, 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 (2022) 58–83.

[11] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, Learning deep features for discriminative localization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[12] Filip Karlo Došilović, Mario Brčić, Nikica Hlupić, Explainable artificial intelligence: A survey, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2018, pp. 0210–0215.

[13] Christoph Molnar, Interpretable Machine Learning, Lulu. com, 2020.

[14] Timo Speith, A review of taxonomies of explainable artificial intelligence (XAI) methods, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 2239–2250.

[15] Gayda Mutahar, Tim Miller, Concept-based explanations using non-negative concept activation vectors and decision tree for CNN models, 2022, arXiv preprint arXiv:2211.10807.

[16] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, Samuel Rota Bulo, Deep neural decision forests, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1467–1475.

[17] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, Joseph E. Gonzalez, NBDT: Neural-backed decision trees, 2020, arXiv:2004.00221.

[18] Joshua Lockhart, Daniele Magazzeni, Manuela Veloso, Learn to explain yourself, when you can: Equipping concept bottleneck models with the ability to abstain on their concept predictions, 2022, arXiv preprint arXiv:2211.11690.

[19] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, Percy Liang, Concept bottleneck models, in: International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348.

[20] Mateo Espinosa Zarlenga, Pietro Barbiero, Zohreh Shams, Dmitry Kazhdan, Umang Bhatt, Mateja Jamnik, On the quality assurance of concept-based representations, 2022, URL https://openreview.net/forum?id=Ehhk6jyas6v.

[21] David Alvarez Melis, Tommi Jaakkola, Towards robust interpretability with self-explaining neural networks, Adv. Neural Inf. Process. Syst. 31 (2018).

[22] Adrien Bennetot, Gianni Franchi, Javier Del Ser, Raja Chatila, Natalia Díaz-Rodríguez, Greybox XAI: a neural-symbolic learning framework to produce interpretable predictions for image classification, Knowl.-Based Syst. 258 (2022) 109947.

[23] Hussein Hazimeh, Natalia Ponomareva, Petros Mol, Zhenyu Tan, Rahul Mazumder, The tree ensemble layer: Differentiability meets conditional computation, in: International Conference on Machine Learning, PMLR, 2020, pp. 4138–4148.

[24] Alberto Lamas, Siham Tabik, Policarpo Cruz, Rosana Montes, Álvaro Martínez-Sevilla, Teresa Cruz, Francisco Herrera, MonuMAI: Dataset, deep learning pipeline and citizen science based app for monumental heritage taxonomy and classification, Neurocomputing 420 (2021) 266–280.

[25] Ivan Donadello, Luciano Serafini, Integration of numeric and symbolic information for semantic image interpretation, Intell. Artif. 10 (1) (2016) 33–47.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[27] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, Alan Yuille, 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, 2014, pp. 1971–1978.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[29] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, 2015, pp. 1–15.

[30] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou, Training data-efficient image transformers and distillation through attention, 2020, http://dx.doi.org/10.48550/ARXIV.2012.12877, URL https://arxiv.org/abs/2012.12877.

[31] Ross Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, ICCV, 2017.

[33] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018, http://dx.doi.org/10.48550/ARXIV.1802.02611, URL https://arxiv.org/abs/1802.02611.

[34] Antonio Gallego Burín, Guía de Granada, Facultad de Letras, 1936.

[35] Tim Miller, Explanation in artificial intelligence: Insights from the social sciences, Artif. Intell. 267 (2019) 1–38.

[36] A.M.M. Sharif Ullah, Khalifa H. Harib, A human-assisted knowledge extraction method for machining operations, Adv. Eng. Inform. 20 (4) (2006) 335–350.

[37] Ashish Kumar, Karan Sehgal, Prerna Garg, Vidhya Kamakshi, Narayanan C Krishnan, MACE: Model agnostic concept extractor for explaining image classification networks, 2020, arXiv:2011.01472.

[38] Andres Felipe Posada Moreno, Nikita Surya, Sebastian Trimpe, ECLAD: Extracting concepts with local aggregated descriptors, 2023, URL https://openreview.net/forum?id=FvqcQ_9u7Mo.

[39] Amirata Ghorbani, James Wexler, James Zou, Been Kim, Towards automatic concept-based explanations, 2019, arXiv:1902.03129.