Explainable Deep Learning Models for Image Classification

UNIVERSIDAD
DE GRANADA

DOCTORAL THESIS

David Morales Rodríguez

PhD Program in Information and Communication Technologies
Department of Electronic and Computer Tecnology
University of Granada

June 2024

Editor: Universidad de Granada. Tesis Doctorales
Autor: David Morales Rodríguez
ISBN: 978-84-1195-444-0
URI: https://hdl.handle.net/10481/94993

Explainable Deep Learning Models for Image Classification

Directed by:

Prof. Diego Pedro Morales Santos
Prof. Manuel Pegalajar Cuellar

PhD Program in Information and Communication Technologies
Department of Electronic and Computer Tecnology
University of Granada

June 2024

Acknowledgements

Muchas gracias a mis directores Manuel y Diego por haberme guiado y apoyado durante esta tesis, y a Estefanía y Beatriz que pusieron la primera piedra y con las que tanto aprendí. Gracias también a todos los compañeros, amigos y en especial a mi familia por haberme acompañado siempre.

Resumen

La inteligencia artificial (IA) ha evolucionado a un ritmo vertiginoso, transformando radicalmente diversos aspectos de nuestra sociedad así como la manera de entender e interactuar con la tecnología. Este desarrollo no podría entenderse sin la visión por computador, que ha sido uno de los grandes campos de pruebas para la maduración de la IA. Muchos modelos y técnicas desarrollados y perfeccionados en este campo han sido luego extrapolados a otras tareas, haciendo que la evolución de la IA fuera en muchas ocasiones ligada o impulsada por la visión por computador. En concreto, gran mérito de esta evolución lo tienen los modelos de aprendizaje profundo que, sin ser sólo propios del campo de la visión por computador, se han visto influenciados por esta. Estos modelos han demostrado una gran versatilidad y precisión en diversos campos y son a día de hoy empleados para resolver multitud de tareas. Sin embargo, una de las grandes críticas que se le hacen a estos modelos es su opacidad.

En esta tesis nos hemos centrado en el campo de la clasificación de imágenes y en estudiar posibles soluciones a este problema de opacidad. En concreto se han estudiado y analizado los retos y limitaciones de estos algoritmos de clasificación con respecto a su interpretabilidad y transparencia.

Como resultado de esta labor de investigación y análisis, se han realizado cuatro propuestas concretas que se presentan en forma de cuatro artículos científicos que componen este documento. Estos artículos proponen respectivamente: 1) la integración de técnicas post-hoc en el entrenamiento de clasificadores; 2) la propuesta de un clasificador entrenado en un entorno multitarea para ser explicable; 3) la propuesta de un modelo explicable basado en conceptos humanos y árboles de decisión difusos; y 4) la mejora del modelo propuesto en el artículo anterior por medio de aprendizaje neurosimbólico y reglas lógicas, para logar un modelo que permita la interacción con el usuario y la incoorporación de conocimiento experto.

En documento se recogen estos artículos con sus resultados y conclusiones, así como la metodología usada y conclusiones finales obtenidas.

Abstract

The artificial intelligence (AI) has evolved at a remarkably fast speed, radically transforming various aspects of our society as well as the way we understand and interact with technology. This development could not be understood without computer vision, which has been one of the major research areas and testing grounds for AI maturity. Many models and techniques developed and optimized in this field have been extrapolated to other research areas, leading the evolution of AI to be often driven by computer vision. Specifically, deep learning models deserve much credit for this evolution, which, while not exclusively associated with the field of computer vision, have been strongly influenced by it. These models have demonstrated great versatility and accuracy in various fields and are currently employed to solve a multitude of different tasks. However, one of the major criticisms leveled against these models is their opacity.

In this thesis, we have focused on the field of image classification and on studying possible solutions to this opacity problem. Specifically, the challenges and limitations of these classification algorithms regarding their interpretability and transparency have been studied and analyzed. As a result of this research and analysis, four specific proposals have been made, presented in the form of four scientific articles comprising this document. These articles propose respectively: 1) the integration of post-hoc techniques in classifier training; 2) the proposal of a classifier trained in a multitask environment to be explainable; 3) the proposal of an explainable model based on human concepts and fuzzy decision trees; and 4) the improvement of the model proposed in the previous article through neurosymbolic learning and logical rules, to achieve a model that allows interaction with the user and the incorporation of expert knowledge.

This document compiles these articles with their results and conclusions, as well as the methodology used and final conclusions obtained.

Contents

List of Figures

List of Tables

Part I

PhD Dissertation

Chapter 1

Introduction

While initial machine learning models were transparent and easy to interpret, recent years have been marked by the surge of less transparent decision-making systems like deep neural networks (DNNs) (2). In the field of computer vision, the potential of convolutional deep learning models has been proven and nowadays these models constitute the state-of-the-art on many tasks due to their great generalization and prediction skills.

However, as said before, these new decision systems based on deep learning can be opaque to the users due to the absence of any mechanism to explain the decision-making process (13). That is why they are considered to be black-box models. In other words, the user feeds the algorithm with an input and gets an output without insight into the reasoning or logic guiding the decision-making process within the algorithm.

This lack of transparency results on a problem in detecting whether the algorithm is biased or whether decisions are being made on reasonable factors. Moreover, users tend to place greater trust in decisions accompanied by explanatory rationales rather than those lacking clarification.

In critical domains such as aviation, medicine, or autonomous driving, where decisions may entail significant risks, it is important that the user can trust the system. Additionally, the transparency and explainability of decisions are crucial factors for industrial verification purposes.

This problem has not only gained the attention of researchers and machine learning experts. The European Commission (EC) already published in 2019 an ethical guideline for Trustworthy AI, defining 7 key requirements that AI-based decision-making systems should meet in order to be deemed trustworthy. The document states, "... AI systems and their decisions should be explained in a manner adapted to the stakeholder concerned" (10).

All this motivated the definition of a new research area to face all these challenges. The goal of Explainable Artificial Intelligence (XAI) is to enhance the transparency of AI-driven systems. Providing human-understandable explanations of the process and outcomes of these systems (especially black-

box models) and allowing users to analyse and understand their behaviour should result on a higher confidence in these systems so that they can be used and approved in certain domains.
There is a necessity for novel XAI methods to comprehend deep learning models. The application of these methods is expected to result in models that are more reliable for users and suitable for industrial validation. In addition, future deep learning models should be constructed with these constraints in mind to enhance their interpretability and comprehensibility. Steps have been taken towards this goal in recent years, but thus far, the quest for explainability frequently compromises algorithm performance.

Computer vision and image classification have been some of the fields where deep learning models have been most widely explored, studied and applied. During this period of predoctoral research, we have focused on the field of XAI applied to image classification and studied potential solutions to the opacity problem of deep learning models in this field. Specifically, we have examined and analyzed the challenges and limitations of these algorithms for image classification in terms of their interpretability and transparency. As a result of this research and analysis, four specific proposals have been made, presented in the form of the four scientific articles comprising this document: 1) the integration of post-hoc techniques in classifier training, 2) the proposal of a classifier trained in a multitasking environment to be explainable, 3) the proposal of an explainable model based on human concepts and soft decision trees, and 4) the improvement of the model proposed in the previous article through neurosymbolic learning and logical rules, to achieve a model that allows interaction with the user and the incorporation of expert knowledge. This document compiles these articles with their results and conclusions, as well as the methodology used and some final conclusions result of these four years of research.

This document is divided in two parts. This first part is organized as follows: in this Chapter 1 we have offered an introduction, in Chapter 2 we present our motivation to start this research period, and summarize the objectives and the methodology. In Chapter 3 we present the main results related to the four articles that form this thesis. The last Chapter 4 of the first part presents the conclusions and future work.

The second Part 5 of this document compiles the four articles that ground this thesis:

- Chapter 6: Integration of post-hoc techniques in classifier training.

- Chapter 7: Proposal of a classifier trained in a multitasking environment to be explainable through image counterfactuals.

- Chapter 8: Proposal of an explainable model based on human concepts and soft decision trees.
- Chapter 9; Improvement of the previous model through neurosymbolic learning and logical rules to allow interaction with the user and the incorporation of expert knowledge.

Chapter 2

Motivation, objectives and methodology

2.1. Motivation

Interpretable deep learning models are increasingly important in domains where transparent decision-making is required. In recent years, there has been a notable increase in contributions to explainable artificial intelligence. Many of these research works had the aim of reducing the opacity inherent in deep learning models.

One of the main criteria to classify explainable techniques and models is to distinguish between intrinsic and post hoc interpretability. In other words, the question is if the models are interpretable by themselves (transparency or intrinsic interpretability) or if explanation methods must be applied after model training to interpret them (3).

In the context of computer vision, post-hoc local explanations, which refer to the use of interpretation methods on single predictions after training a model have been widely explored and used.

Most of the best known post-hoc techniques provide heat maps to identify the regions of the input images that networks look at when making predictions, allowing the data to be interpreted at a glance such as Local Interpretable Model-agnostic Explanations (LIME) technique (11) or GradCAM (12). The use of counterfactuals describing a causal situation in the form: “If X had not occurred, Y would not have occurred” has also gained attention. In the context of image classification, given an input data point and a model, a counterfactual is defined as a generated data point that is as close to the input data point as possible but for which the model gives a different outcome (9).

However, heatmaps maps or altered images generated by these post-hoc explanation techniques may not be sufficient to understand the decision-making process of the model. Furthermore, the following consideration must be made: consider a scenario where a model exhibits a confidence level of 95 %, and an independent post-hoc explanation technique yields a confidence level of 95 % for the same decision. In such a case, the combined confidence in the decision, given both the model's output and the post-hoc explanation, would be reduced to 90.25 %.

On the other hand, the creation of transparent high-perfomance models stands as one of the main goals of XAI, representing an ongoing research area. The main challenge lies in the historical trade-off between the interpretability/transparency of the models and their accuracy (5).

In recent years some research has been dedicated to resolve this well-known trade-off challenge. Some promising trends are the fusion of classical algorithms with the power of deep learning methods (6), the use of concept learning (1) or the use of neural-symbolic learning (4). The study of the human intervention of the models and the user-model interaction is also an interesting field of research (7, 8). In this direction, some authors such as Miller (8) emphasized the nature of explanations as a form of knowledge transfer resulting from interactions.

In summary, post-hoc techniques have been widely explored, however the definition of explainable models and the reduction of the historical trade-off in XAI is still an active area of research. Furthermore, we consider the exploration of interpretable solutions enabling user interaction a relatively underexplored research area. We believe that the interaction of the user with the model is a condition for interpretability and to gain the user's trust.

All this formed the basis of our motivation to start a research on explainable deep learning models applied to image classification.

2.2. Objectives

Confronting the historical trade-off between transparency and performance, the main objective of this doctoral thesis is to develop a deeper understanding of computer vision techniques and models with the aim of contributing to make them more interpretable and transparent without compromising accuracy, which should lead to increasing user trust on deep learning models. The objectives of this research period can be specified as follows:

1. The first goal involves conducting an exhaustive analysis of the literature in the field of explainable computer vision, focusing on image classification. The aim is to gain expert knowledge and to understand the current trends and limitations. State-of-the-art techniques and models should be identified together with their strengths and weaknesses. The knowledge and expertise acquired in this phase should form a solid background, enabling the development of subsequent research objecti-

ves.

2. To explore the use of known explanation techniques with the aim of improving the performance of the models, as well as the development of new explanation and interpretability techniques. The limitations found in the previous analysis could result in interesting research opportunities. Furthermore, one of our hypothesis is that the use of explanation during the learning process could improve the robustness and performance of the models. The new models based on the current techniques may be adapted to this purpose.

3. Development of more transparent architectures and models. These models should be interpretable without needing of any post-hoc explanation methods. This has been always one of the main challenges in XAI due to the historical trade-off between the interpretability/transparency of the models and their accuracy.

4. To explore the interaction of models and users as a way of improving the interpretability of the models and in order to gain the user's trust. Providing users with control over the decision-making process may enhance their trust on the models. The incorporation of user knowledge as expert knowledge may be a step in this way. Bringing the human into the loop of the learning process would also enable the user to adapt the model to his necessities.

2.3. Methodology

In this section, we discuss the methods implemented to achieve the goals outlined in Section 2.2.

- Analysis and observation: review of the state-of-the-art and understanding of current trends, methods and limitations. The aim is to gain knowledge and expertise in the field of focus and to identify research opportunities.

- Hypothesis development: design of new techniques, models and approaches to face and solve the previously identified problems and limitations. The objectives defined in previous Sections should motivate this phase. Original approaches should be proposed based on the knowledge acquired during the previous research phase.

- Hypothesis test: selection of performance metrics and comparison procedures to test the proposed approaches and to compare them to representative approaches in the literature. The hypothesis shall be compared to the current state-of-the-art methods to decide if the results obtained achieve the proposed objectives.

- Thesis extraction: the final phase of this research work involves the redaction of a document explaining the conclusions and results obtained as well as the processes and methods. This document is the result of this phase.

Chapter 3

Results

This thesis is found in a compendium of publications. This compilation consists of three articles indexed in three different scientific journals, complemented by an additional article currently under review in a fourth journal. These publications present the most relevant results achieved during the doctoral research period. In this chapter we summarize the main results related to the four articles that form this thesis.

- **David Morales, Estefania Talavera and Beatriz Remeseiro, *Playing to distraction: towards a robust training of CNN classifiers through visual explanation techniques,* in Neural Computing and Applications, 2021. DOI: 10.1007/s00521-021-06282-2.**

  In this article, we introduced a novel training approach that incorporates visual explanation techniques into the learning process. Our training proposal is designed to enhance the robustness and generalization capability of Convolutional Neural Networks (CNNs) in image classification tasks. The aim is to intervene in the learning process by forcing the model to focus not only on relevant regions but also on those that, a priori, are not so informative for the discrimination of the class. During the training phase, a visual explanation algorithm is used to identify the areas on which the model bases its decisions. These areas are occluded and the model is trained with a combination of the modified images and the original images. The proposed approach was tested by embedding it into the learning process of several state-of-the-art convolutional neural network architectures on two challenge datasets. Additionally, the proposed solution was tested on a real-case scenario for the classification of egocentric images. Furthermore, diverse robust experiments were performed. The results obtained in all the performed experiments demonstrate that the proposed training approach contributes to improve the robustness of the model and its generalization capabilities.

- **David Morales, M. P. Cuellar and D. P. Morales, *On the fusion of Soft-Decision-Trees and Concept-based models* in Applied Soft Computing, 2024, DOI: 10.1016/j.asoc.2024.111632.**

  In this research work, the focus lies on the use of decision trees in combination with deep learning models. With the aim of improving the transparency of the the decision-making process for image classification, we explore how to combine and train decision trees and concept-based models. The proposed approach consists of an interpretable image classifier that bases its decision on human-understandable concepts. The final classification is conducted by a soft decision tree. This tree is transparent and can be visualized and explored by the user. We discussed and analysis our model as intepretable model and demonstrated that the proposed approach opens the door to human intervention. An expert user could explore and analyse the decision tree, being able to improve the model by using his knowledge to redefine and modify the decision paths.

- **David Morales, M. P. Cuellar and D. P. Morales, *Concept logic trees: enabling user interaction for transparent image classification and human-in-the-loop learning,* in Applied Intelligence, 2024, DOI: 10.1007/s10489-024-05321-4.**

  In this research, we introduced a novel approach that integrates soft decision trees, neural symbolic learning, and concept learning to construct an image classification model. The proposed solution enhances interpretability and facilitates user interaction, control, and intervention. The fusion of an interpretable architecture with neural symbolic learning allows the incorporation of expert knowledge and the interaction of the user with the model. Additionally, the proposed solution enables the inspection and analysis of the model through queries in the form of first-order logic predicates. This all results in a human-in-the-loop transparent model. The experimental results on challenging datasets validate the proposed approach, demonstrating a competitive performance when compared to state-of-the-art solutions.

- **David Morales, M.P. Cuellar and D. P. Morales, *Exploring methods for the generation of visual counterfactuals in the latent space* UNDER REVIEW.**

  In this research work, we investigated how to train a model in a multitasking environment with both tasks in mind: classification and visual counterfactual explanation. In other words, we propose a self-explaining image classifier so that it can produce its own counterfactuals. Our proposed architecture is based on variational autoencoders, where we define a conditional generator to generate visual counterfactuals by modifying the latent representations. In our method, the classifier and counterfactual model are trained together, sharing the same latent space, which enhances interpretability.

Chapter 4

Conclusions

This doctoral research has focused on the exploration of explainability in the field of image classification and the development of more transparent models and techniques. In this Chapter, I first summarized in Section the conclusions extracted from each of the articles that found this thesis. Furthermore, in Section I present some personal conclusions that are the result of these years of predoctoral research.

4.1. Conclusions extracted from each research work

We summarize the main conclusions that we extracted of each research work below:

- **Playing to distraction**: In our first article, after studying post-hoc explanation techniques, we demonstrated their potential to enhance the performance of image classification models. We introduced a novel training approach that improves the generalization ability and robustness of CNNs applied to image classification. Our training approach was based on applying a visual explanation algorithm to identify the areas on which the model bases its decisions. After identifying those areas, we occluded them and trained the model with a combination of the modified images and the original ones. Our method forces the network to learn additional features that help it distinguish between very similar classes, showing its suitability for fine-grained classification problems. We demonstrated the adequacy of our training scheme regardless of the backbone architecture considered. We conducted a series of occlusion and visual explanation experiments, validating that our approach enhances classifier robustness by forcing the model to rely on a broader range of features during the decision-making process.

  We believe that applying our proposed algorithm not only to the input images but also to the feature maps obtained at different convolutional levels is an interesting future work. Similar as done with the regularization technique known as dropout which is usually applied at different architecture levels, our algorithm could force the model to pay attention to different characteristics on the feature maps, thereby improving the robustness of the model at different levels.

- **Exploring methods for the generation of visual counterfactuals in the latent space:** In this research work, we focused on defining and training an interpretable convolutional image classifier preparing it for the generation of image counterfactuals. In our proposed approach we define an architecture where the encoder, classifier, and decoder share a common latent space. This enables us to analyse this latent space and to use it to generate counterfactuals. Our aim is to advance towards self-explanatory techniques, diverging from purely post-hoc approaches. Our proposed solution achieves competitive results when compared with state-of-the-art methods.

  In future research, we aim to further explore the latent space of various models, including generative models, believing that a rich latent space can enhance interpretability and self-explanation. We see potential in transferring knowledge between tasks by sharing layers or latent spaces, or by training models in multitasking environments. Our goal is to delve deeper into this research domain to develop more interpretable architectures and models. While the models in this study were custom-made, we anticipate leveraging pre-defined architectures for classification and generation to tackle more intricate datasets. Specifically, we're interested in investigating style GANs, known for their rich and disentangled latent spaces. Finally, addressing the use of protoypes in dataset with high intra-class variation is identified as one challenging task. Our current prototype-based solution shares this limitation, and we intend to explore extensions to overcome this challenge in future research.

- **On the fusion of Soft-Decision-Trees and Concept-based models:** In this study, we investigate the integration of decision trees with deep learning models. We develop an interpretable classification model enabling classification based on human-understandable concepts. This is accomplished through an architecture founded on Concept Bottlenecks and Soft-Decision-Trees. By employing soft-decision trees, we can train the models using gradient-based optimization methods, as known from classical deep-learning models. We experiment with various methods of multitasking training, forcing the model to utilize human-labeled concepts for final classification. As a result, we present an interpretable architecture providing transparent decision-making for users. Comparative analysis with state-of-the-art methods demonstrates competitive performance, all achieved without requiring object detectors or object-part annotations.

The main critic to models based on concept learning is that most of them requires prior annotation of concepts, and this applies also to our proposed approach. We believe that the combination of our model with automatic concept extraction methods is an interesting future task. Furthermore, we believe that by combining our work with other techniques such as rule extraction mechanisms or pruning methods could lead to make our model more transparent and optimize it. Finally, our proposal opens the door to human intervention. An expert is able to explore the model and could even improve or custimize the decision-making process by modifying the decision paths. We believe that further research must be conducted in this direction of human intervention and interaction in order to enhance user's trust and improve the transparency of deep learning models.

- **Concept logic trees:**

  In this article, our aim was to create an interpretable classification model that relies on human-understandable concepts and allows for user intervention and the integration of expert knowledge. The proposed solution is based the fusion of soft decision trees, neural symbolic learning, and concept learning. The proposed approach offers the ability to impose constraints on the routing process of the soft decision tree using first-order logic rules and predicates based on concept or class combinations. This gives users greater control over decision-making, determining which nodes or leaves to explore for specific classes or in response to particular concepts. This level of control makes the model highly adaptable and customizable to suit specific needs and preferences. By defining rules and predicates, users can intervene during training and inspect the model's reasoning process. The resulting concept-based architecture is interpretable, incorporates expert knowledge, and facilitates user control and intervention. We evaluated our approach on two challenging datasets, comparing it with state-of-the-art solutions. Our method achieved competitive results and outperformed existing state-of-the-art transparent models on the PASCAL dataset.

  We believe that continue exploring the potential of combining neural symbolic learning and soft decision trees is a promising research line. This combination could enhance model interpretability, transparency, and adaptability, thereby serving as a robust tool for decision-making not only in image classification but also in other domains like reinforcement learning, where soft decision trees are extensively studied. Our proposed solution, like other supervised concept learning models, relies on concept annotations. To address this, some researchers have explored methods such as automatic concept extraction, and we see potential in combining these approaches with our own. Lastly, we believe there's a need for continued research to enhance the interaction between models and humans in deep learning, aiming to increase trust in AI systems. We believe that neural symbolic learning could play an important role on facing this last challenge.

4.2. Thesis conclusions

As a result of all the work realized during the past four years, I would like to add some personal conclusions on XAI and image classification:

- After studying post-hoc techniques in the first two articles, I believe that these can be useful as a complement for improving performance, identifying biases, or as analyzing tools, but not as sole XAI solutions. AI models should be transparent and interpretable by design and architecture, not interpretable only afterward.

- The definition of what an interpretable model is should depend on the user and his context. What may be understandable for user A may not suffice as an explanation for user B. Furthermore, enabling user interaction with the model should be crucial for a model to be considered transparent.

- Just like in any other field of software engineering, the use of large models inherently makes models less transparent. The same applies in our domain. Employing the divide and conquer technique is a strategy applicable in explainable computer vision as well. Breaking down tasks into smaller subtasks performed by smaller models instead of applying one large model to a single task can enhance interpretability.

- AI will suppose or is supposing a revolution in our lives, which is a source of fear for many people. Similar to many other previous revolutions in the past, this fear is mostly associated to a lack of understanding or incomprenhension. One of the main challenges of AI in general and XAI in particular is to alleviate this fear.

Chapter 5

References

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[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. *Information Fusion*, 58:82–115, 2020.

[3] Diogo Vieira Carvalho, Eduardo Marques Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 2019. URL `https://api.semanticscholar.org/CorpusID:199659548`.

[4] Natalia Díaz-Rodríguez, Alberto Lamas, Jules Sanchez, Gianni Franchi, Ivan Donadello, Siham Tabik, David Filliat, Policarpo Cruz, Rosana Montes, and Francisco Herrera. Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. *Information Fusion*, 79:58–83, 2022.

[5] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

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

[7] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[8] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[9] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[10] High-Level Expert Group on AI. Ethics guidelines for trustworthy ai. Technical report, European Commission, 08 April 2019. URL `https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai`.

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

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

[13] Warren J von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

Part II

Publications

Chapter 6

Playing to distraction: towards a robust training of CNN classifiers through visual explanation techniques

David Morales[1,2], Estefanía Talavera[3], Beatriz Remeseiro[4].

1. Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
2. HAT.tec GmbH. c/o Universit¨at der Bundeswehr, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany.
3. Department of Computer Science, University of Groningen. Nijenborgh 9, 9747 AG Groningen, Netherlands.
4. Department of Computer Science, Universidad de Oviedo. Campus de Gijón s/n, 33203 Gijón, Spain.

Neural Computing and Applications

- Received: 28 December 2020 / Accepted: 26 June 2021 / Published online: 9 July 2021.
- DOI: 10.1007/s00521-021-06282-2
- Impact factor: 5.10
- JCR Rank: 45/145 in category COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (Q2)

Abstract:

The field of deep learning is evolving in different directions, with still the need for more efficient training strategies. In this work, we present a novel and robust training scheme that integrates visual explanation techniques in the learning process. Unlike the attention mechanisms that focus on the relevant parts of images, we aim to improve the robustness of the model by making it pay attention to other regions as well. Broadly speaking, the idea is to *distract* the classifier in the learning process by forcing it to focus not only on relevant regions but also on those that, *a priori*, are not so informative for the discrimination of the class. We tested the proposed approach by embedding it into the learning process of a convolutional neural network for the analysis and classification of two well-known datasets, namely Stanford cars and FGVC-Aircraft. Furthermore, we evaluated our model on a real-case scenario for the classification of egocentric images, allowing us to obtain relevant information about people's lifestyles. In particular, we work on the challenging EgoFoodPlaces dataset, achieving state-of-the-art results with a lower level of complexity. The results obtained indicate the suitability of our proposed training scheme for image classification, improving the robustness of the final model.

6.1. Introduction

Nowadays, the potential of convolutional deep learning models for the task of image classification has been proven. Research in this field has followed different directions namely, new architecture and framework proposals (27, 19), training methods (37, 38), multi-tasking (40, 24), attention mechanisms (23, 18), explainability and interpretability (28, 36), among others.

New techniques such as attention mechanisms allow to force the model to pay attention to certain features, whilst explainable artificial intelligence techniques allow to interpret the model and know what is happening during the learning process. However, to the best of our knowledge, the combination of both approaches has not been explored. Inspired by this lack of combination, we aim to improve the training procedure by interpreting the model and focusing it on certain regions of interest. To this end, our proposed approach is based on modifying the classical training procedure to include online information and thus adapt the learning process based on the features on which the network is focused.

More specifically, we propose a new training scheme that benefits from the saliency maps provided by visual explanation techniques. Our hypothesis is that, by the end of the training phase, the model should use as many features as possible to make a robust prediction. In this sense, we apply a visual explanation algorithm to identify the regions on which the model bases its decisions. After identifying those relevant areas, we partially occlude them trying to *distract* the model in some way and forcing the detection of other regions that, a priori, are weak (i.e., not so informative for the discrimination of the class). Our intention is to highlight that the model should not forget what the occluded regions mean, but it should learn to recognize other features to make a decision. This is ensured as the occluded images are combined with the original ones during the learning process.

We think fine-grained image classification problems could benefit the most from this approach, as they have many classes that differ from each other in small details, and our training approach forces the network to find them. For this reason, we evaluated the proposed training scheme on two well-know datasets namely Stanford cars (22) and FGVC-Aircraft (25), composed of 16,185 and 10,000 images respectively, and used in fine-grained recognition. In addition, we carried out some experiments on top of different backbone architectures to demonstrate that our proposal improves the performance regardless of the respective network.

Furthermore, we evaluate the robustness of our model in a real-scenario case study: recognizing the food-related scene that an egocentric image depicts. The analysis of egocentric images is an emerging field within computer vision that has gained attention in recent years (9). Images captured by wearable cameras during daily life allow recording information about the lifestyle of the users from a first-person perspective (5, 34). The analysis of this information can be used to improve peoples'health-related habits (13). In particular, the analysis of food-related egocentric images can be a powerful tool to analyze peoplesñutritional habits, being the focus of previous research (29, 34). In this context, we carried out some experiments on the EgoFoodPlaces dataset (34), which is composed of 33,801 images and describes food-related locations gathered by 11 camera wearers throughout their daily life activities.

The contributions of this research work are three-fold:

1. A novel training scheme for CNN image classification that makes use of visual explanation techniques, with the main aim of improving the robustness and the generalization ability of the trained models.

2. The experiments carried out demonstrate the competitiveness of our training scheme, which outperforms the classical approach on two public datasets commonly used in fine-grained recognition tasks, regardless of the backbone architecture.

3. Our proposed method achieves competitive results in a real-case scenario that addresses the classification of egocentric photo-streams depicting food-related scenes.

The rest of the paper is organized as follows. Section 9.2 includes an overview of related works. Section 9.3 presents the proposed training approach. Section 6.4 introduces the two datasets for fine-grained recognition, describes the experiments carried out and analyzes the obtained results. Section 6.5 describes and evaluates the case study focused on egocentric vision. Finally, Section 9.5 closes with our conclusions and future lines of research.

6.2. Related Work

While the very first machine learning systems were easily interpretable, the last years have been characterized by an upsurge of opaque decision systems, such as deep neural networks (DNNs) (3, 4). DNNs are the state-of-the-art on many machine learning tasks due to their great generalization and prediction skills. However, they are considered *black-box* machine learning models. In this context, there has been a growing influx of work on explainable artificial intelligence. Post-hoc local explanations, which refer to the use of interpretation methods after training a model, and feature relevance methods are increasingly the most adopted approaches to explain DNNs (3). In this section, we review some methods that produce *visual explanations* for decisions of a large class of DNN-based models, making them more transparent and reliable.

Most of these visual explanation techniques provide heat maps to identify the regions of the input images that networks look at when making predictions, allowing the data to be interpreted at a glance. Note that these heat maps are also referred to in the literature as sensitivity maps, saliency maps, or class activation maps. Class activation mapping (CAM) (42) is a well-known procedure for generating class activation maps using global average pooling in CNNs. Their authors expect each unit to be activated by some visual pattern within its receptive field. The class activation map is nothing more than a weighted linear sum of the presence of these visual patterns at different spatial locations. By simply upsampling the class activation map to the size of the input image, they can analyze the most relevant image regions to identify the particular category. However, CAM can only be used with a restricted set of layers and architectures.

A class-discriminative localization technique called gradient-weighted class activation mapping (Grad-CAM) was proposed in (31). In fact, it is a generalization of CAM that can be applied to a significantly broader range of CNN families. Grad-CAM uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map, highlighting the regions of the image that are relevant for the prediction. Given an image and a class of interest (e.g., *tiger cat*) as inputs, Grad-CAM forward propagates the image through the convolutional part of the model and then through task-specific computations to obtain a raw score for the category. The gradients are set to 0 for all classes except for the desired class (*tiger cat*), which is set to 1. This signal is then backpropagated to the rectified convolutional feature maps of interest, which are combined to compute the coarse Grad-CAM localization that represents where the model looks at to make the corresponding decision. Finally, they point-wise multiply the heat map with guided backpropagation, thus obtaining also guided Grad-CAM visualizations, which are both high-resolution and concept-specific.

Another visual explanation method was presented in (39), in which input images are perturbed by occluding all their patches, in an iterative process, and classifying the occluded images. This idea allows the authors to analyze how the top feature maps and the classifier output change, revealing structures within each patch that stimulate a particular feature map. However, the use of this method requires generating multiple occluded samples and their classification, making it computationally expensive.

Ribeiro et al. (26) proposed the local interpretable model-agnostic explanations (LIME) technique, which allows to explain the predictions of any classifier in an interpretable and faithful manner. Given the original representation of the instance being explained, they get new samples by perturbing the original representation. They use those samples to approximate the classifier with an interpretable model. Just as the method above, the use of multiple samples implies to apply the classifier several times given one instance.

Some of these visual explanation techniques generate noisy sensitivity maps. In this context, Smilkov et al. (32) proposed SmoothGrad, a technique to reduce the noise in the sensitivity maps produced by visual explanation techniques based on gradients. Their idea was to sample images similar to the original ones by adding some noise. Then, they produced intermediate sensitivity maps for each image and took the average of them as the final sensitivity map.

Finally, it is worth highlighting some applications of the saliency maps generated by visual explanation techniques. Schöttl (30) used Grad-CAM maps to improve the explainability of classification networks. More specifically, the idea was to introduce some measures obtained from the Grad-CAM maps in the loss function. Cancela et al. (6) proposed a saliency-based feature selection method that selects the features that contain a higher discrimination result, allowing to provide robust and explainable predictions in both classification and regression problems.

6.2.1. Egocentric photo-streams

Following, we review some recent works on egocentric photo-streams, mainly focused on the classification of food-related scenes, such as our case study.

Egocentric image analysis is a field within computer vision related to the design and development of algorithms to analyze and understand photo-streams captured by wearable cameras (34). These cameras are capable of capturing images that record visual information of our daily life, known as *visual lifelogging*, to create a visual diary with activities of first-person life. The analysis of these egocentric photo-streams can improve people's lifestyle by analyzing social patterns (15), social interactions (2), or food behavior (35).

In recent years, there is a growing interest in egocentric photo-streams giving their potential for assisted living. For instance, Furnari et al. (12) presented a benchmark dataset containing egocentric videos of eight personal locations and proposed a multi-class classifier to reject locations not belonging to any of the categories of interest for the end-user.

As for food-related scene recognition, Sarker et al. (29) addressed this task by proposing a multi-scale atrous CNN (7) to analyze lifelogging images and determine people's recurrences in food places throughout their day. Later, Talavera et al. (34) presented the EgoFoodPlaces dataset, composed of more than 33,000 images organized in 15 food-related scene classes. This dataset was recorded by 11 users while spending time on the acquisition, preparation, or consumption of food. The dataset was manually labeled into a total of 15 different food-related scene classes like *bakery shop*, *bar*, or *kitchen*. Taking into account the relation of the studied classes, a taxonomy for food-related scene recognition was introduced. Furthermore, the authors proposed a hierarchical classification model based on the aggregation of six VGG16 networks (20) over different subgroups of classes, emulating the proposed taxonomy. This is, to the best of our knowledge, the state-of-the-art in the recognition of food-related scenes in egocentric images.

6.3. Methodology

We propose a novel training approach to improve the robustness of CNNs in image classification. Figure 6.1 illustrates the different steps of the proposed scheme, which are subsequently explained in depth.

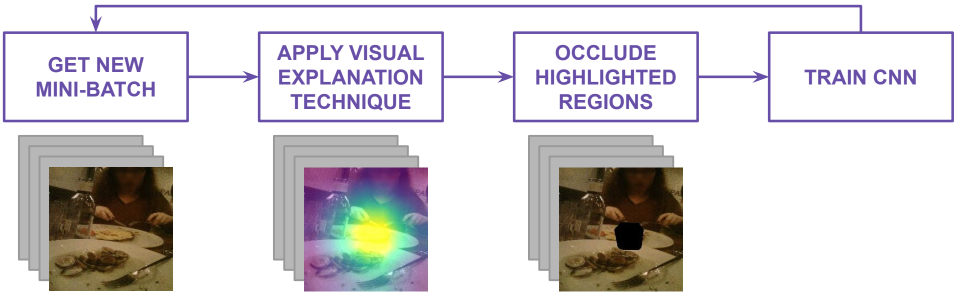
GET NEW MINI-BATCH
APPLY VISUAL EXPLANATION TECHNIQUE
OCCLUDE HIGHLIGHTED REGIONS
TRAIN CNN

Figure 6.1: Workflow of our alternative training scheme, which (1) gets a new mini-batch of input images, (2) applies a visual explanation technique to generate the heat maps, (3) occludes the regions highlighted in the previous step, and (4) trains the CNN classifier.

Let consider the classical mini-batch gradient descent (10) training algorithm where, on each training step, the mini-batch is first fed into the neural network, then the gradient is computed, and finally, the calculated gradient is used to update the weights of the network. We propose to modify the training step to apply the new scheme over each mini-batch with a probability $p \in (0, 1)$; i.e., with a probability $1 - p$, the images in the mini-batch kept unchanged and the classical training step is performed as usual. Note that the probability p belongs to the open interval $(0, 1)$. $p = 0$ would mean that our training scheme is not applied (i.e., the classical training procedure is used instead). $p = 1$ would mean that only the modified images are used, making model convergence difficult. Preliminary experimentation suggests applying the method with values of $p \leq 0{,}5$ to guarantee that both occluded and original images are used in the learning process. Therefore, with a probability $p \in (0, 1)$, our training scheme is applied as follows:

1. Using the current weights of the network, we do inference over the current mini-batch and apply a visual explanation method to get a heat map for each image in the mini-batch. These heat maps highlight the regions where the current model focuses its attention to classify the corresponding image.

2. After that, we occlude the areas corresponding to those highlighted regions, forcing the model to look at other regions in the image. For each image in the mini-batch, we normalize its heat map and get a weight $w \in [0, 1]$ for each pixel. Next, we select all the pixels whose weight w is over a threshold th. The selected pixels are erased by setting them to 0, calling this approach 0-occlusion. As a result, we obtain the occluded images of the mini-batch.

3. Finally, we train our model making use of the occluded mini-batch.

Algorithm 2 shows the pseudo-code of our proposed training method according to the 0-occlusion approach. Note that once the mini-batch is modified, the training step continues as usual (i.e., the gradient is calculated and the weights are updated). We think it is important to highlight that the model should not forget what the occluded regions mean, but learn to recognize other parts of the image to make a decision. This is guaranteed as the occluded images are used only for some mini-batches, according to the p hyper-parameter, while the original ones are used for the rest of them.

The proposed approach is compatible with any of the visual explanation methods presented in Section 9.2 and, in general, with any method that generates a heat map to explain the decision of a CNN for a given target image. Among all these techniques, we choose Grad-CAM (31) because it uses the flow of the gradients from the last convolutional layer to compute the heat maps, making it computationally less expensive than other methods

```
Data: trainingSet, model, $p$, $th$
Result: the model trained using the proposed approach
for miniBatch in trainingSet do
    $r$ = random(0,1);
    if $r \leq p$ then
        for (image, label) in miniBatch do
            heatMap = visualExplanation(image, label, model,
              lastConvLayer);
            heatMap = minMaxNorm(heatMap);
            selectedPixels = [heatMap > $th$];
            image[selectedPixels] = 0;
        end
    end
    train(model, miniBatch);
end
```

Algorithm 1: Pseudo-code of the proposed training scheme using 0-occlusion.

like LIME (26) or SmoothGrad (32). These other techniques apply inference several times on images generated by perturbing the target image to compute the heat maps. In other words, Grad-CAM does inference once per image while other techniques do inference several times per image, which makes the former more appropriate for the problem at hand.

Summarizing, the heat maps provided by Grad-CAM highlight the relevant regions of the image for predicting the ground truth class. By occluding them, the model is forced to look at other regions to make the decision. The initial regions should not be forgotten by the model, but other parts of the images should also be taken into account in the learning process. In this manner, the model improves its robustness and generalization capabilities.

6.4. Experimental framework 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 two experiments carried out, including the evaluation metrics considered. Finally, we report and analyze the results obtained in both experiments: (1) a comparison between the proposed method and some variants of it, and (2) a comparison with standardized baselines.

6.4.1. Datasets

We evaluated our proposed method on two well-known datasets: the Stanford cars dataset (22), and the fine-grained visual classification of aircraft (FGVC-

Aircraft) benchmark dataset (25). Both datasets were used as part of the fine-grained recognition challenge FGComp 2013, which ran jointly with the ImageNet Challenge 2013[1].

The Stanford cars dataset contains 16,185 images of 196 car models covering sedans, SUVs, coupes, convertibles, pickups, hatchbacks, and station wagons; and it is officially split into 8,144 training and 8,041 test images. The FGVC-Aircraft dataset contains 10,000 images of aircraft, with 100 images for each of 100 different aircraft model variants; and it is officially split into 6,667 training and 3,333 test images.

6.4.2. Implementation details

The techniques and parameters used for experimentation are explained in the following. We used the Adam optimization algorithm (21) with the following parameters: learning rate $\alpha = 0{,}00005$, $\beta_1 = 0{,}9$, $\beta_2 = 0{,}999$, and $\epsilon = 0{,}0000007$. Regarding the training step, we used a batch size of 16 and the images were resized to 224×224. The outputs were monitored using the validation accuracy to apply an early stopping strategy, based on which the training process finished after 30 epochs with no improvement. Additionally, we applied the following classical data augmentation techniques: horizontal flip, rotation $[-40, 40]$, random channel shift $[-30, 30]$, and image brightness change $[0{,}5, 1{,}5]$.

The proposed method was implemented on TensorFlow (1) and Keras (8), and the code is available for download[2]. Starting from the training algorithm provided in Keras, we modified the training step to apply our method over each mini-batch with a probability p, as described in Section 9.3. According to some preliminary experiments, we applied the proposed method with a probability $p = 0{,}25$, and the threshold for the occlusion step was set to $th = 0{,}85$.

6.4.3. Experimental setup

This section describes the two experiments designed to evaluate our training scheme. Both experiments were applied to each dataset individually and compared with other approaches. As for the experimentation itself, we kept the original split in training and test sets for the two considered datasets (see Section 9.4.1). For validation purposes, we randomly divided the original training dataset into two parts: 75 % training and 25 % validation. Then, we trained the model and evaluated it on the isolated test set, using the performance metrics described in Section 9.4.4. This validation procedure was repeated five times. We report the average performance and the standard deviation calculated across the five runs.

[1] http://image-net.org/challenges/LSVRC/2013/

[2] https://github.com/DavidMrd/Playing-to-distraction

6.4.3.1. Experiment 1

The objective of this experiment is to test several setups of our training scheme and compare them with a baseline. In particular, we used a ResNet50 (14), a very popular network successfully applied to different image classification tasks. The different configurations are detailed as follows:

1. **Baseline.** In order to compare our method with a baseline, we trained a ResNet50 using the classical training approach (i.e., without applying the proposed method). We called this model fine-tuned ResNet50 (FT-ResNet50) because it is a model pre-trained on the ImageNet dataset (11), whose parameters were fine-tuned with the corresponding dataset.

2. **Our approach.** We trained a ResNet50 using the proposed training method, which is based on Grad-CAM visualizations and illustrated in Figure 6.1. More specifically, we used the weights from the ResNet50 model pre-trained on ImageNet (11), and then we fine-tuned them using the corresponding dataset and our training scheme. Note that, during the learning process, the Grad-CAM algorithm was applied to the last convolutional layer of the ResNet50, as indicated in (31).

3. **Other setups.** Aiming at demonstrating the adequacy of the 0-occlusion approach, we also conducted some experiments in which the pixels were set to a random value (R-occlusion) and 1 (1-occlusion).

6.4.3.2. Experiment 2

This experiment aims to demonstrate the adequacy of our training scheme regardless of the backbone architecture considered. In this sense, we applied it to two well-known backbone architectures, different from ResNet50, using the following configurations:

1. **Baselines.** We trained two architectures commonly used in the literature, InceptionV3 (33) and DenseNet (17), using the classical approach. We called them FT-InceptionV3 and FT-DenseNet, respectively, because they were pre-trained on ImageNet and fine-tuned with the corresponding dataset.

2. **Our approach.** We trained the two backbone architectures considered, InceptionV3 and DenseNet, using the proposed training scheme (see Figure 6.1). As in the previous experiment, we used the weights from these two architectures pre-trained on ImageNet, and then we fine-tuned them with the corresponding dataset and our training scheme. Regarding the Grad-CAM algorithm, it was applied to the last convolutional layer of the networks as described in (31).

6.4.4. Evaluation

In order to evaluate the performance of the proposed models and make a fair comparison with other approaches, we computed some popular metrics in image classification tasks: accuracy, precision, recall, and F-score (F1). These metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.4}$$

where TP, FP, TN, and FN stand for true positives, false positives, true negatives, and false negatives, respectively.

6.4.5. Results

In this section, we report and analyze the results obtained in the two experiments described above.

6.4.5.1. Experiment 1

Table 6.1 shows the results obtained for the different configurations. As can be observed, our training scheme provides very competitive results regardless of the setup used for the occlusion. Analyzing the four metrics considered, the three setups outperform the baseline method (FT-ResNet50), which was trained with the classical learning procedure, in both datasets. Focusing on our proposal (0-occlusion), it achieves a gain of more than 2 percent in the Standford cars dataset and about 2 percent in the FGVC-Aircraft dataset. In order to demonstrate the relevance of this improvement, we applied a statistical t-test that allows us to determine if there is a significant difference between the baseline (FT-ResNet50) and our proposal (0-occlusion). Notice that we used a paired sample, two-tailed t-test. As a result, we can confirm that our proposal significantly outperforms the baseline in terms of accuracy, with a significance level of 0.05.

If we analyze the behavior of the three different setups considered for the proposed training scheme, we can see that both 0-occlusion and 1-occlusion provide better results than R-occlusion, with a very slight difference in favor of the former (0-occlusion). The experiments show that, when using random values for the occlusion procedure, the model does not benefit so much from the *distraction* applied to the model, by forcing it to look at new regions in the input images. This behavior is discussed in detail below, with some qualitative results that aim at illustrating the impact of the proposed method.

Stanford cars				
	FT-ResNet50	0-occlusion	R-occlusion	1-occlusion
Accuracy	0,849 ± 0,009	**0,871 ± 0,007**	0,860 ± 0,009	0,869 ± 0,008
Precision	0,855 ± 0,007	**0,876 ± 0,007**	0,866 ± 0,008	0,873 ± 0,008
Recall	0,849 ± 0,009	**0,870 ± 0,008**	0,860 ± 0,009	0,868 ± 0,009
F1	0,848 ± 0,009	**0,870 ± 0,008**	0,859 ± 0,009	0,867 ± 0,009
FGVC-Aircraft				
	FT-ResNet50	0-occlusion	R-occlusion	1-occlusion
Accuracy	0,731 ± 0,013	**0,749 ± 0,005**	0,739 ± 0,012	0,743 ± 0,005
Precision	0,746 ± 0,011	**0,762 ± 0,005**	0,755 ± 0,010	0,759 ± 0,004
Recall	0,731 ± 0,013	**0,749 ± 0,005**	0,739 ± 0,012	0,743 ± 0,005
F1	0,731 ± 0,014	**0,748 ± 0,005**	0,739 ± 0,012	0,743 ± 0,005

Table 6.1: Classification performance, averaged across five runs, of the different approaches on the Stanford cars (22) and FGVC-Aircraft (25) datasets. Best results are in bold.

Figure 6.2 depicts two representative images of the two datasets used for experimentation, Stanford cars and FGVC-Aircraft, along with the heat maps generated by Grad-CAM for the different methods analyzed: the baseline FT-ResNet50 and the three setups for the proposed training approach. As can be observed, the models trained with the proposed approach, regardless of the setup, base their decisions on more features than the one trained using a classical approach (FT-ResNet50). While the baseline method seems to base its decisions just on a local area of the image, the models trained with the proposed approach seem to react to almost the whole object. Analyzing the different configurations, we can see that both 0-occlusion and 1-occlusion show a similar behavior, reacting to the whole object, which explains the achieved results in both cases. However, the R-occlusion version behaves differently since it reacts to many features but with a low level of confidence. That is, occluding the selected pixels with a fixed value (0 or 1) allows us to achieve better results than occluding the relevant regions with a random value. The reason for this behavior could be that, when using a fixed value, the model learns to ignore these areas and looks at other regions, whereas the model does not benefit as much from this idea when using a different value each time. It is worth noting that using 0-occlusion is somewhat similar to the well-known dropout (16), a regularization technique in which some connections are disabled during the learning phase. This would explain why this approach gets slightly better results than the 1-occlusion version.

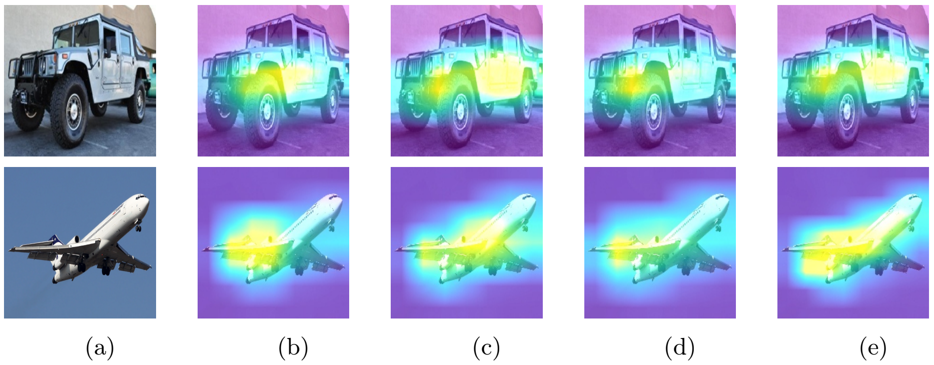

Figure 6.2: (a) Input images from the Stanford cars (top) and FGVC-Aircraft (bottom) datasets, (b) heat maps generated by Grad-CAM for the baseline FT-ResNet50, and heat maps generated by Grad-CAM for the model trained with the proposed training scheme using (c) 0-occlusion, (d) R-occlusion, and (e) 1-occlusion.

Finally, Table 6.2 shows the number of epochs and the seconds per epoch needed to train the baseline (FT-ResNet50) and our proposal (0-occlusion). As can be observed, our training scheme requires more computational time per epoch and more epochs to converge than the classical procedure. Regarding the increment in terms of seconds per epoch, it is lower than 19 %. Note that this time only depends on the image resolution and the hardware, so it is the same for both datasets. With respect to the increment in the number of epochs, it is ≈ 32 % for the Stanford cars dataset and ≈ 53 % for the FGVC-Aircraft dataset. Nevertheless, it is worth noting that, for application purposes, this computational time is not decisive since the training procedure is carried out only once before the model is put into production, after defining its architecture and setting its hyper-parameters. As our method is applied during the learning process, the computation time in the test phase is not affected.

	FT-ResNet50		0-occlusion	
	Standford-C	FGVC-A	Standford-C	FGVC-A
Nº epochs	98,8 ± 6,78	113,4 ± 10,97	130 ± 10,38	174 ± 13,87
s per epoch*	153 ± 0		175 ± 0	

Table 6.2: Number of epochs and seconds per epoch, averaged across five runs, needed to train the two different approaches on the Stanford Cars (22) and FGVC-Aircraft (25) datasets. * Network input size: 224 × 224 × 3. Hardware: NVIDIA T4 Tensor Core GPU.

6.4.5.2. Experiment 2

Table 6.3 shows the results obtained when applying our training scheme to the other two backbone architectures selected: InceptionV3 and DenseNet. According to the figures, our approach outperforms the corresponding baseline for both datasets regardless of backbone considered. While analyzing the behavior of our training scheme when using InceptionV3, we can observe that it achieves an improvement of more than 1 percent for the four performance measures. In terms of accuracy, this improvement over the baseline is of 1.3 percent on the Stanford cars dataset and 1.5 percent on the FGVC-Aircraft dataset. Regarding the DenseNet backbone, the improvement with respect to the baseline is about 1.1 percent for all the metrics on both datasets.

Stanford cars				
	FT-Inception	0-occl-Inception	FT-DenseNet	0-occl-DenseNet
Acc	0,778 ± 0,023	**0,791 ± 0,020**	0,883 ± 0,010	**0,894 ± 0,011**
Prec.	0,788 ± 0,021	**0,798 ± 0,020**	0,888 ± 0,009	**0,898 ± 0,011**
Rec.	0,777 ± 0,023	**0,791 ± 0,020**	0,882 ± 0,010	**0,893 ± 0,012**
F1	0,776 ± 0,023	**0,790 ± 0,021**	0,882 ± 0,010	**0,893 ± 0,012**
FGVC-Aircraft				
	FT-Inception	0-occl-Inception	FT-DenseNet	0-occl-DenseNet
Acc	0,618 ± 0,029	**0,633 ± 0,026**	0,767 ± 0,026	**0,780 ± 0,025**
Prec	0,630 ± 0,030	**0,641 ± 0,029**	0,786 ± 0,024	**0,794 ± 0,023**
Rec	0,618 ± 0,028	**0,633 ± 0,026**	0,767 ± 0,026	**0,780 ± 0,025**
F1	0,616 ± 0,029	**0,630 ± 0,026**	0,768 ± 0,026	**0,780 ± 0,025**

Table 6.3: Classification performance (Accuracy, Precision, Recall and F1-Score) averaged across five runs, making use of different backbones on the Stanford cars (22) and FGVC-Aircraft (25) datasets. Best results are in bold.

6.5. Case study

This section describes an application of the proposed method to a real-world scenario. In particular, we consider the task of food-related scene classification in egocentric images, as detailed below.

6.5.1. Dataset

We evaluated our proposed method on the EgoFoodPlaces dataset (34), which is composed of 33,810 egocentric images gathered by 11 users and organized in 15 food-related scene classes. By making use of a wearable camera[3], the users regularly recorded an amount of approximately 1,000

[3] http://getnarrative.com/

images per day. The camera movements and the wide range of different situations that the users experienced during their days, lead to challenges such as background scene variation or changes in lighting conditions. The dataset was manually labeled into a total of 15 different food-related scene classes namely, *bakery shop, bar, beer hall, cafeteria, coffee shop, dining room, food court, ice cream parlor, kitchen, market indoor, market outdoor, picnic area, pub indoor, restaurant, and supermarket.* Table 6.4 depicts the distribution of images among the collected classes, with a great imbalance between them.

Class	Bakery shop	Bar	Beer hall	Cafeteria	Coffee shop	Dining room	Food court	Ice cream parlor	Kitchen	Market indoor	Market outdoor	Picnic area	Pub indoor	Restaurant	Supermarket	Total
#Images	144	1632	672	1689	2313	3639	204	107	3837	1181	1388	921	511	10310	5262	**33810**

Table 6.4: Distribution of images per class in the EgoFoodPlaces dataset (34).

6.5.2. Experimental results

This section describes the results obtained when addressing the task of food-related scene classification with our proposed training scheme.

The implementation details are the ones described in Section 9.4.2 with two exceptions: (1) the resolution of the input images, which in this case is 250 × 250 as in (34); and (2) the application of class oversampling to the fourth largest class (i.e., *dining room*) in order to alleviate the imbalance problem.

Concerning the experimentation, we used the split described in (34), which includes a division into events for the training and evaluation phases, to make sure that there are no images from the same scene/event in both phases. The validation procedure, in this case, consisted of three partitions, with the following distribution: training set (70 %), validation set (10 %), and test set (20 %). Then, the model was trained and evaluated on the test set. This validation procedure was repeated five times. We report the average performance and the standard deviation calculated across the five runs.

Finally, we considered the four performance metrics detailed in Section 9.4.4: accuracy, precision, recall, and F1 score. Note that, for the per-class metrics (precision, recall, and F1), we calculated the macro- and weighted-averages, as suggested in (34): *macro* gives equal weight to all classes, while *weighted* is sensitive to imbalances. It is worth noting the relevance of these two average values due to the unbalanced nature of the dataset.

6.5.2.1. Classification performance

For the evaluation of our proposal, we followed the experimental setup described in Section 6.4.3, but using the EgoFoodPlaces dataset to train the ResNet50 architecture with the classical procedure (FT-ResNet50) and with our training scheme (0-occlusion). Additionally, we compared our results with the ones reported in (34), the state-of-the-art approach for this dataset.
Table 6.5 reports the results obtained for the different approaches. As can be seen, training a ResNet50 with our proposed scheme (0-occlusion) allows us to achieve a higher accuracy than the one obtained with the baseline (FT-ResNet50). Moreover, the proposed method also achieves higher weighted averages for the other three metrics considered (precision, recall, and F1). It is worth noting that, due to the high imbalance of the dataset, the weighted metrics are more informative than the macro values. Concerning the latter, the differences between both methods are minimal, with the same values for precision and F1, and a slightly higher macro recall in favor of the baseline.

	Hierarchical (34)	FT-ResNet50	0-occlusion
Macro Precision	0.56	**0,59 ± 0,03**	**0,59 ± 0,05**
Macro Recall	0.53	**0,55 ± 0,03**	0,54 ± 0,06
Macro F1	**0.53**	**0,53 ± 0,04**	**0,53 ± 0,06**
Weighted Precision	0.65	0,67 ± 0,02	**0,68 ± 0,03**
Weighted Recall	**0.68**	0,67 ± 0,03	**0,68 ± 0,04**
Weighted F1	0.65	0,64 ± 0,03	**0,66 ± 0,04**
Accuracy	**0.68**	0,67 ± 0,03	**0,68 ± 0,04**

Table 6.5: Classification performance, averaged across five runs, of the different approaches on the EgoFoodPlaces dataset (34). Best results are in bold.

If we analyze the results achieved by the state-of-the-art (34) and compare them with the proposed method, we can see that our approach achieves better results in four out of the seven performance measures, whereas the remaining three are equal. We find important to point out that our approach makes use of only just one classifier (ResNet50), while the model presented in (34) uses a hierarchical ensemble composed of six VGG16 networks. Therefore, the complexity of our model is significantly lower, not only because we have one single classifier but also because our backbone model ResNet50 has a lower number of parameters than their VGG16 networks. Therefore, we can conclude that our proposed method is able to achieve similar performance results with a less complex architecture and a computationally less expensive approach.
Finally, the impact of the different approaches on the individual classes is presented in Table 6.6. As can be seen, our method (0-occlusion) shows a behavior very similar to the baseline approach (FT-ResNet50), with slightly

higher rates in seven classes and three ties. Analyzing the figures obtained with the hierarchical approach (34), our method achieves better results in eight classes. More specifically, the results in which our approach outperforms the state-of-the-art correspond to the four most represented classes (*restaurant*, *supermarket*, *kitchen*, and *dining room*). Also noteworthy is the improvement achieved for the class *food court*, which could not be classified by the hierarchical model (true positive rate of 0.00). However, there are five classes for which the hierarchical model gets a better performance, including *beer hall*, *cafeteria*, and *coffee shop*. We deduce that this is due to the benefits of classifying them in a hierarchical fashion.

Class	Bakery shop	Bar	Beer hall	Cafeteria	Coffee shop	Dining room	Food court	Ice cream parlor	Kitchen	Market indoor	Market outdoor	Picnic area	Pub indoor	Restaurant	Supermarket
Hierarchical (34)	0.39	0.31	**0.89**	**0.45**	**0.59**	0.58	0.00	**0.52**	0.89	**0.70**	0.28	0.00	**0.85**	0.70	0.85
FT-ResNet50	**0.63**	0.31	0.24	0.38	0.49	0.66	**0.53**	0.50	0.89	0.60	**0.57**	0.00	0.78	**0.73**	0.90
0-occlusion	0.60	**0.32**	0.26	0.35	0.48	**0.72**	0.43	**0.52**	**0.90**	0.60	0.53	0.00	0.80	**0.73**	**0.92**

Table 6.6: True positive rate per class, averaged across five runs, of the different approaches on the EgoFoodPlaces dataset (34). Best results are in bold.

Going deeper into the results obtained and given the characteristics of the EgoFoodPlaces dataset, we can draw some additional conclusions. Firstly, we can observe that the classification improves when using our approach for (1) classes where the scene to recognize is right in front of the camera users (e.g., *restaurant*), and (2) classes that tend to share descriptors even if recorded at different locations (e.g., *dining room* or *supermarket*). Those results inherit that the model is able to learn the relevant features in the scene when it is self-contained, which is closely related to the fine-grained datasets evaluated in Section 6.4.

Analyzing the images we can also see that, in some classes (e.g., *food court*, *cafeteria*, *market outdoor*), there is more background than foreground information necessary for the identification of the scene; that is, the image is composed of characteristics that an observer would not find relevant for the distinction of an event. Therefore, the main difficulty in learning these scenes is that not only the locations vary but also they are composed of elements common to other scenes. In this case, including other relevant regions along with a limited amount of samples available per class might represent imposed noise and lead to a lower performance in our approach compared to the baseline. This issue could be addressed with the extension of the dataset.

6.5.2.2. Model inspection

We analyzed not only the classification performance of our training scheme but also its ability to make predictions. In particular, we aimed to find out if the proposed approach is able to improve the robustness of a CNN classifier and make it sensible to more features. For this reason, we carried out two additional experiments: (1) we analyzed the behavior of the models making use of a visual explanation algorithm, and (2) we randomly erased some areas of the test images before evaluating the models on them.

In the first experiment, our target was to demonstrate that the regions considered as relevant by the trained models were more and bigger when applying our training scheme than when following the classical procedure. For this purpose, we applied the Grad-CAM algorithm to the last convolutional layer of the two ResNet50 models previously trained on the EgoFoodPlaces dataset: one trained using the classical procedure (FT-ResNet50), and the other one using our training scheme with 0-occlusion. As a result, we obtained the heat maps that allow us to visualize the regions that are important to the models when making a prediction for a given image. Figure 6.3 depicts some representative images along with their corresponding heat maps for each model. As can be observed, our model took into account bigger regions than the baseline method (FT-ResNet50) when processing the same target images. Besides, it can be seen that the model trained with our proposed method bases its decisions on more regions than when using the classical procedure. Furthermore, the regions that the baseline model took into account when making a decision were also taken into account by the proposed model. This demonstrates that when using the proposed training scheme, the model does not forget the learned features, but just learns to recognize other features.

Finally, we conducted the second experiment to test the robustness of our training scheme. For this purpose, we hid some regions of the test images by randomly erasing them, as proposed in (41). After that, we compared how the two approaches (FT-ResNet50 and 0-occlusion) performed on the modified test set. Table 6.7 presents the results for this experiment. As can be observed, the proposed approach (0-occlusion) performs better than the baseline model (FT-ResNet50). This means that our model does not suffer as much when some areas of the image are erased or hidden, demonstrating its robustness. It is also worth noting that these results are consistent with the ones obtained in the previous experiment, and demonstrate that our model makes use of more and bigger regions than the baseline approach to make a prediction for a target image.

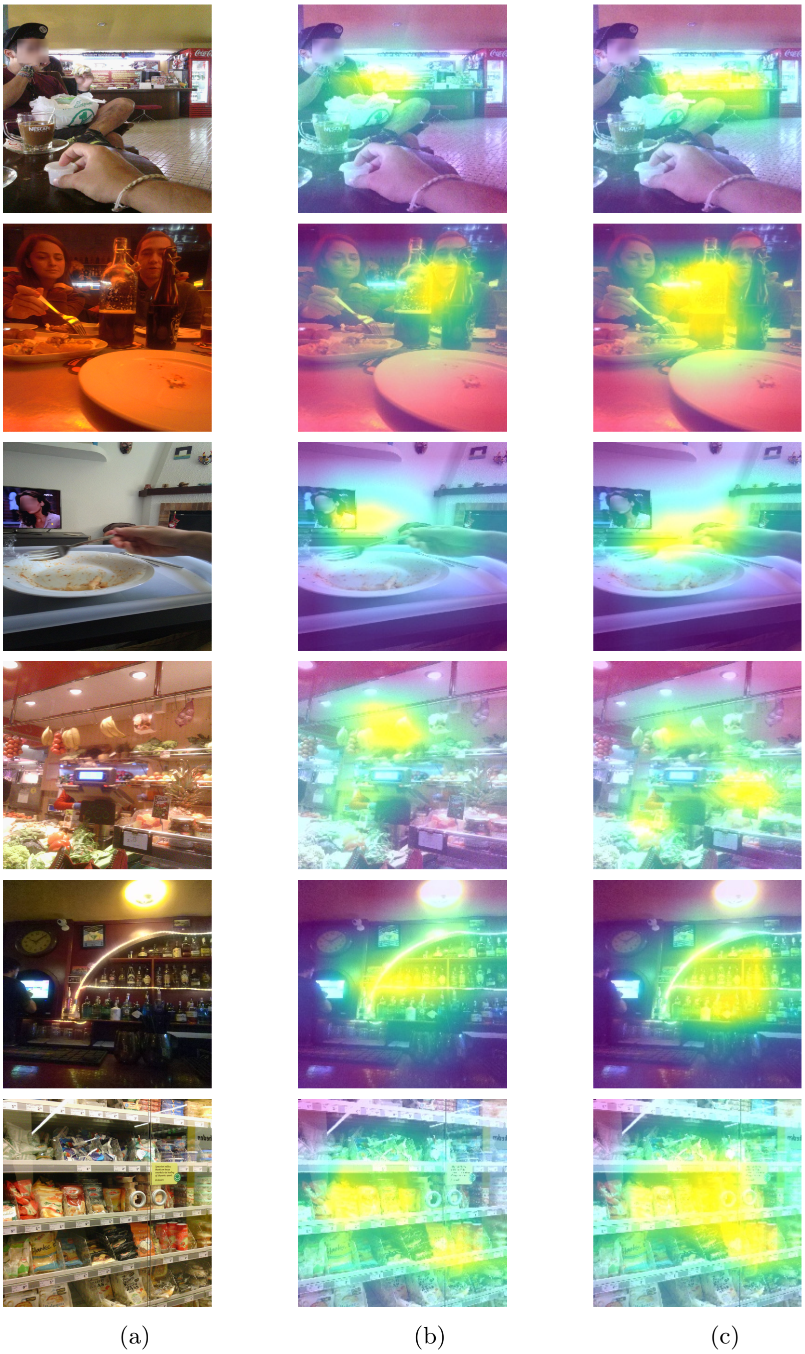

Figure 6.3: (a) Input images, (b) heat maps generated by Grad-CAM for the baseline FT-ResNet50, and (c) heat maps generated by Grad-CAM for the model trained with the proposed training scheme (0-occlusion).

	FT-ResNet50	0-occlusion
Macro Precision	0,53 ± 0,01	**0,54 ± 0,02**
Macro Recall	0,47 ± 0,02	**0,48 ± 0,03**
Macro F1	0,47 ± 0,02	**0,48 ± 0,05**
Weighted Precision	**0,63 ± 0,02**	**0,63 ± 0,03**
Weighted Recall	0,59 ± 0,02	**0,65 ± 0,03**
Weighted F1	**0,59 ± 0,02**	**0,59 ± 0,02**
Accuracy	0,59 ± 0,02	**0,60 ± 0,02**

Table 6.7: Classification performance, averaged across five runs, of the baseline method and the proposed training scheme when we randomly hid some regions on the test images. Best results are in bold.

6.6. Conclusion

This research work presents a novel training scheme that improves the robustness and generalization ability of CNNs applied to image classification. The idea is to force the model to learn as many features as possible when making a class selection. For this purpose, we apply a visual explanation algorithm to identify the areas on which the model bases its decisions. After identifying those areas, we occluded them and trained the model with a combination of the modified images and the original ones. In this manner, the model is not able to base its prediction on the occluded regions and is forced to use other areas. Consequently, the model also learns to pay attention to those regions of the target image that, *a priori*, are not so informative for its classification.

To evaluate the proposed method, we carried out different experiments on two popular datasets used for fine-grained recognition tasks: Stanford cars and FGVC-Aircraft. From the obtained results, we can confirm our initial hypothesis: our method forces the network to learn additional features that help it distinguish between very similar classes, showing its suitability for fine-grained classification problems. More specifically, and within the different evaluated configurations, the 0-occlusion approach has shown to be the most appropriate setting. Furthermore, we demonstrated the adequacy of our training scheme regardless of the backbone architecture considered.

We further experimented with a real-case study focused on the classification of food-related scenes. We analyzed the impact of our training scheme by comparing it with a baseline method and, to the best of our knowledge, with the state-of-the-art approach that follows an ensemble composed of six CNNs (34). The results achieved with our method were comparable or even better than the ones obtained with the state-of-the art approach despite making use of just one network, thus reducing the level of complexity while maintaining a competitive performance. Furthermore, our method is computationally less expensive, as the chosen backbone (ResNet50) has fewer parameters than the VGG16 used in (34). Finally, we carried out several occlusion and visual explanation experiments, showing that our method improves the robustness of the classifier by forcing it to base its decisions on more features.

As a future line of research, it would be interesting to apply the same methodology not only to input images but also at different convolutional levels, as it is usually done with the regularization technique known as dropout. In other words, the feature maps obtained at different levels could be analyzed and occluded in the same way that we did with the input images. This idea would force the model to pay attention to different characteristics on the feature maps, thereby improving the robustness of the model at different levels of the learning process.

6.7. acknowledgements

We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. *USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[2] Maedeh Aghaei, Mariella Dimiccoli, and Petia Radeva. With whom do I interact? Detecting social interactions in egocentric photo-streams. *International Conference on Pattern Recognition*, pages 2959–2964, 2016.

[3] 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. *Information Fusion*, 58:82–115, 2020.

[4] Adrien Bennetot, Jean-Luc Laurent, Raja Chatila, and Natalia Díaz-Rodríguez. Towards explainable neural-symbolic visual reasoning. *IJCAI Neural-Symbolic Learning and Reasoning Workshop*, 2019.

[5] Marc Bolanos, Mariella Dimiccoli, and Petia Radeva. Toward storytelling from visual lifelogging: An overview. *IEEE Transactions on Human-Machine Systems*, 47(1):77–90, 2016.

[6] Brais Cancela, Verónica Bolón-Canedo, Amparo Alonso-Betanzos, and João Gama. A scalable saliency-based feature selection method with instance-level information. *Knowledge-Based Systems*, 192:105326, 2020.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[8] François Chollet et al. Keras. `https://keras.io`, 2015.

[9] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. *European Conference on Computer Vision*, pages 720–736, 2018.

[10] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13:165–202, 2012.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[12] Antonino Furnari, Giovanni Maria Farinella, and Sebastiano Battiato. Temporal segmentation of egocentric videos to highlight personal locations of interest. *European Conference on Computer Vision*, pages 474–489, 2016.

[13] Olga Gelonch, Neus Cano, Marta Vancells, Marc Bolaños, Laia Farràs-Permanyer, and Maite Garolera. The effects of exposure to recent autobiographical events on declarative memory in amnestic mild cognitive impairment: A preliminary pilot study. *Current Alzheimer Research*, 17 (2):158–167, 2020.

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

[15] Pedro Herruzo, Laura Portell, Alberto Soto, and Beatriz Remeseiro. Analyzing first-person stories based on socializing, eating and sedentary

patterns. *International Conference on Image Analysis and Processing*, pages 109–119, 2017.

[16] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[17] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[18] Deepak Kumar Jain, Rachna Jain, Yash Upadhyay, Abhishek Kathuria, and Xiangyuan Lan. Deep refinement: capsule network with attention mechanism-based system for text classification. pages 1839–1856, 2020.

[19] SanaUllah Khan, Naveed Islam, Zahoor Jan, Ikram Ud Din, and Joel JP C Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, 125:1–6, 2019.

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, pages 1–15, 2015.

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. *4th International IEEE Workshop on 3D Representation and Recognition*, pages 554–561, 2013.

[23] Xiang Li, Wei Zhang, and Qian Ding. Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal Processing*, 161:136–154, 2019.

[24] Diogo C Luvizon, David Picard, and Hedi Tabia. 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.

[25] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

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

[27] Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature Neuroscience*, 22(11):1761–1770, 2019.

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

[29] Mostafa Kamal Sarker, Hatem A Rashwan, Estefania Talavera, Syeda Furruka Banu, Petia Radeva, Domenec Puig, et al. MACNet: Multiscale Atrous Convolution Networks for Food Places Classification in Egocentric Photo-streams. *European Conference on Computer Vision Workshops*, pages 1–11, 2018.

[30] Alfred Schöttl. A light-weight method to foster the (Grad) CAM interpretability and explainability of classification networks. *International Conference on Advanced Computer Information Technologies*, pages 348–351, 2020.

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

[32] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[34] Estefanía Talavera, Maria Leyva-Vallina, Md Mostafa Kamal Sarker, Domenec Puig, Nicolai Petkov, and Petia Radeva. Hierarchical approach to classify food scenes in egocentric photo-streams. *IEEE Journal of Biomedical and Health Informatics*, 24(3):866–877, 2019.

[35] Estefania Talavera, Andreea Glavan, Alina Matei, and Petia Radeva. Eating Habits Discovery in Egocentric Photo-streams. *arXiv preprint arXiv:2009.07646*, 2020.

[36] Alfredo Vellido. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural Computing and Applications*, pages 1–15, 2019.

[37] Juan Wu, Seabyuk Shin, Cheong-Gil Kim, and Shin-Dug Kim. Effective lazy training method for deep q-network in obstacle avoidance and path planning. *IEEE International Conference on Systems, Man, and Cybernetics*, pages 1799–1804, 2017.

[38] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A semantic loss function for deep learning with symbolic knowledge. *International Conference on Machine Learning*, pages 5502–5511, 2018.

[39] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *European Conference on Computer Vision*, pages 818–833, 2014.

[40] Kunpeng Zhang, Liang Zheng, Zijian Liu, and Ning Jia. A deep learning based multitask model for network-wide traffic speed prediction. *Neurocomputing*, 396:438–450, 2020.

[41] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random Erasing Data Augmentation. *AAAI Conference on Artificial Intelligence*, pages 13001–13008, 2020.

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

Chapter 7

On the fusion of Soft-Decision-Trees and Concept-based models

David Morales[1,2], M.P. Cuellar[2], Diego P. Morales[3].

1. HAT.tec, Lilienthalstraße 15, 85579 Neubiberg, Germany
2. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
3. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

Applied Soft Computing

- Received April 2021, Accepted June 2021, Published June 2021
- DOI: 10.1016/j.asoc.2024.111632
- Impact factor: 8.7
- JCR Rank: 21/145 in category COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (Q1)

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.

7.1. 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 (36), 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, 15). 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 (30), which perturbs the input and demonstrates how the predictions change, or Grad-CAM (31), 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 (8). 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 (12, 29, 33, 8). 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.
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-desicion 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 9.2 includes an overview of the state of the art. Section 9.3 presents the proposed methods and training approaches. Section 9.4 introduces the two datasets, describes the experiments carried out, and analyzes the obtained results. Finally, Section 9.5 closes with conclusions and future lines of research.

7.2. 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, 7). 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 can not get an explanation of the decisions made by the model (36). 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) (39) or Local Interpretable Model-Agnostic Explanations (LIME) (30).

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 (10). 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 (26, 32) 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 (28). 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. (20) 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. (37) 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 (12) 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 (24). 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, 38). A well-known article in this field was presented by Alvarez Melis and Jaakkola (1), 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 colours and shapes of objects, which are precise and accurate for human understanding (7). 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 (7, 3).

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

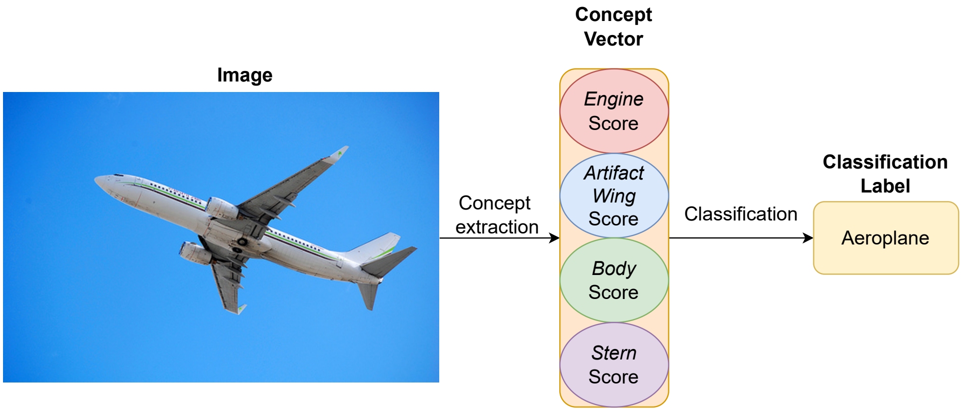

Figure 7.1: Diagram showing a classification task resolved using concept learning. The task is divided into two subtasts: first, a concept extraction takes place producing a concept vector as output. This extraction can be implemented as a multilabel classification problem, where the labels depend on the datasets. The concept vector contains the scores for each label. Second, a classification process takes place based on the concept vector.

tree.

7.3. 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 Figure 7.1. A Soft Decision Tree is used as a predictor, which performs the final classification based on the extracted concepts.

7.3.1. Concept Bottlenecks

The classification problem is divided into two subtasks: concept extraction and classification (see Figure 7.1). The concept extraction task is defined as a multilabel classification problem. The labels (concepts) depend on the dataset, which should be annotated accordingly. After 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 $g : \mathbb{R}^d \rightarrow [0,1]^k$ maps from input space to concept space and $t : [0,1]^k \rightarrow \mathbb{Y}$ 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 $L_Y : \mathbb{Y} \times \mathbb{Y} :\rightarrow \mathbb{R}_+$

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.

7.3.2. 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 (16). 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 (12). 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 (12, 16).

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:

$$p_i(x) = \sigma(xw_i + b_i) \tag{7.1}$$

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 figure 7.2 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

$$P^l(x) = \prod_N p_i(x)^{\mathbf{1}[l \swarrow i]}(1 - p_i(x))^{\mathbf{1}[i \searrow l]} \tag{7.2}$$

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

$$Q^l = softmax(\phi^l) \tag{7.3}$$

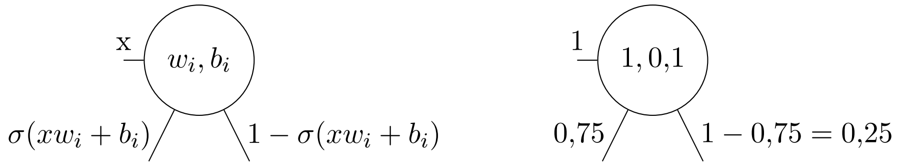

Figure 7.2: On the left side we present an inner node i of a binary soft decision tree. Each inner node has two learned parameters associated: the weight w_i and a bias b_i. On the right side we illustrate the routing process according to Equation 7.1 with an example. To this aim the variables are given the following values: $x = 1, w_i = 1, b_i = 0{,}1$.

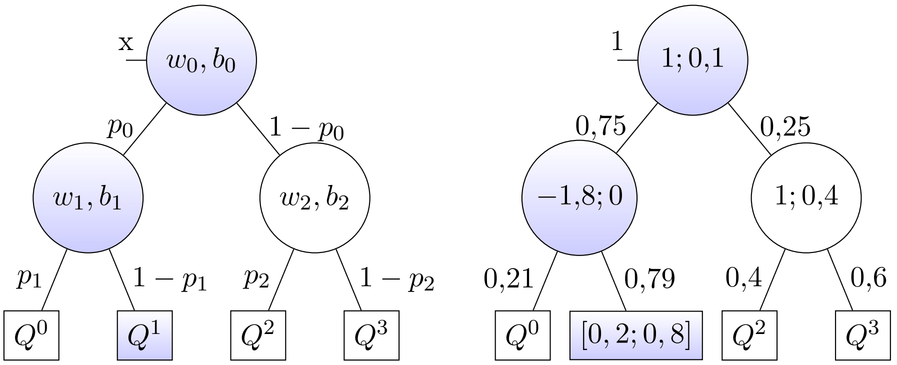

Figure 7.3: On the left figure we show a soft decision tree with one hidden layer for a binary classification problem. In the figure on the right, we assume some values for the input x and for the weights w_i, biases b_i, and learning variables like $Q^1 = [0, 2; 0{,}8]$. The probabilities of routing to the left or to the right are shown for each level.

where ϕ^l is a learned parameter. The output of the model is the distribution at the leaf with the maximum path probability. In Figure 7.3 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 Figure 7.2. 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.

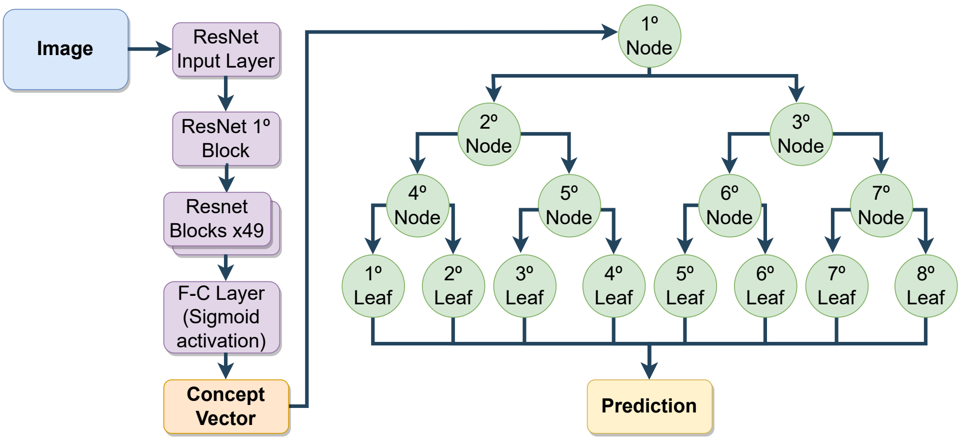

Figure 7.4: Overall structure of the proposed model. The concept extractor g is implemented by a Resnet-50. Its final layer is implemented by a fully connected layer with a sigmoid activation function. The concept extractor gets an image as input and outputs the concept vector. The binary soft decision 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.

The decision tree is trained using the loss function

$$L_T(x) = -\sum_{l \in L} P^l(x) \sum_{k \in Y} y_k \log Q_k^l \tag{7.4}$$

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 Figure 7.3, 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 $L_{l_1}(x) = P^1(x)\sum_{k\in Y} y_k \log Q_k^1 = 0{,}59\ (0 \log 0{,}2 + 1 \log 0{,}8) = -0{,}057$. 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 Equation 8.3.

7.3.3. Overall Structure

In figure 8.2 we show the overall structure of the proposed model.

7.3.4. 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 the subsection 8.3.1 while L_Y correspond to the loss function described in equation 8.3. 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.

7.4. 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.

7.4.1. Datasets

We evaluated the proposed methods on the MonuMAI dataset (22) and on the Semantic PASCAL-Part dataset (9).
The MonuMAI dataset (22) 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 (3, 22).

The PASCAL VOC 2010 dataset (11) is a well-known image dataset organized into 20 object classes. The PASCAL-Part dataset (6) 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. (7) and based on the Semantic PASCAL VOC dataset (9). 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 colour 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 (7, 3).

7.4.2. Implementation details

The proposed method was implemented on Pytorch, and the code is available for download[1]. We use a Resnet-50 (17) 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 (12). After a preliminary analysis, we decided to set the depth parameter of the soft decision trees to 5. We used the Adam optimization algorithm (18) for all networks.

7.4.3. Experiments

This section describes the experiments designed to evaluate the proposed methods (see section 9.3). 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 (3) and (7) for the two considered

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

[2] github-decisiontree

datasets (see Section 9.4.1). 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.

7.4.4. Results

In this section, we report and analyze the results obtained in the experiments described in Section 9.4.3. 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 9.3, 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

$$\text{Y-Acc} = Acc(y, y') \tag{7.5}$$

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

$$\text{C-Error} = 1 - \frac{avg(BinaryAcc(c_i, c_i{}'))_{i \in 1..N}}{100} \tag{7.6}$$

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 7.1 for the MonuMAI dataset and in Table 7.2 for the PASCAL dataset.

MonuMAI			
	Ind-Tree	Seq-Tree	Joint-Tree
Y-Acc	92,38 ± 0,41	92,74 ± 0,21	**97,82 ± 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 7.1: Results of the proposed experiments on the MonuMAI dataset. Best results for the final classification task are in bold.

It can be observed that the Independent (Ind) models and the Sequential (Seq) models performed very similarly on both datasets. Please note that

PASCAL			
	Ind-Tree	Seq-Tree	Joint-Tree
Y-Acc C-Error	84,29 ± 0,39 0,028 ± 0,001	84,36 ± 0,34 0,028 ± 0,001	**85,14 ± 0,53** 0,029 ± 0,001
	Ind-Baseline	Seq-Baseline	Joint-Baseline
Y-Acc C-Error	84,26 ± 0,5 0,028 ± 0,001	84,26 ± 0,39 0,028 ± 0,001	83,06 ± 0,92 0,035 ± 0,01

Table 7.2: Results of the proposed experiments on the PASCAL dataset. Best results for the final classification task are in bold.

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 9.3. 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 simular 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 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 7.6.

7.4.5. 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 9.2. Two of the models are transparent models (Greybox (3) and EXPLANet (7)) and the other two models are opaque models (DeiT-B (34, 3) and MonuNet (22)). 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 8.5. On the MonuMai dataset, the Joint approach achieves higher accuracy than the second-best approach (DeiT-B (34, 3)). 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 (7) requires a complex architecture such as Faster R-CNN (14) or RetinaNet (23), which also increases the complexity of the training. The same issue occurs when a segmentation model such as DeepLab-V3+ (5) is needed, as for Greybox (3). In fact, note that a model based on DeepLab-V3 has necessary more than 101 layers, as ResNet-101 (17) is used as backbone, while our model has less than 60 layers.

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 (3)	94,04	88,30
EXPLANet (7)	90,40	82,4
DeiT-B (34, 3)	96,48	**90,85**
MonuNet (22)	83,11	-

Table 7.3: 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.

7.4.6. Discussion

7.4.6.1. Visualization

In Figure 7.5 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 colours represent negative values and the light colours 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 analysed by using post-hoc explanation methods such as Grad-CAM(31) or LIME (30). In Figure 7.5, we demonstrate this option generating a saliency map for the concept "Broken pediment" which is present in the concept vector.

7.4.6.2. The model as explainable AI model

In order to discuss our proposed approach as explainable AI model, we refer to Miller (25) 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

[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.

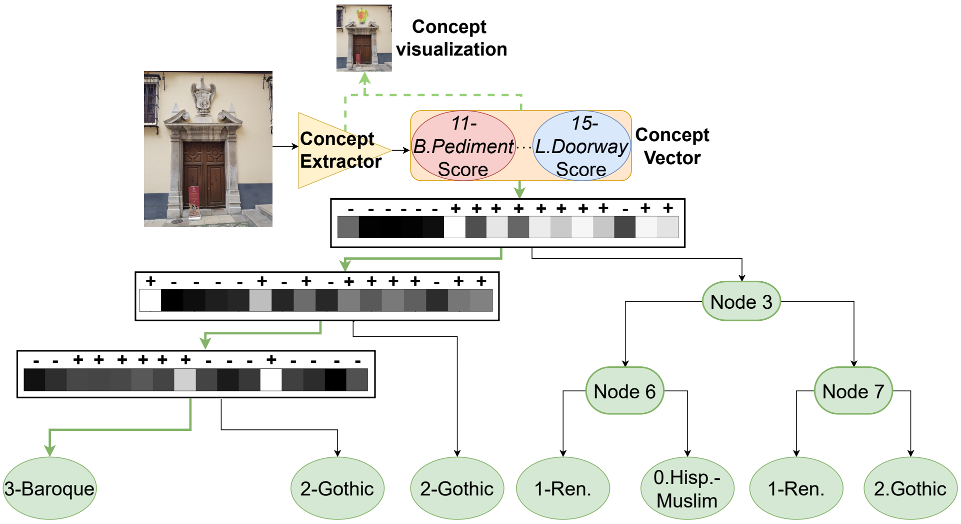

Figure 7.5: Visualization of the prediction process. The green arrows indicate the path through the decision tree during the prediction process for the input x. Sample x is an image taken from the MonuMAI dataset (22). We visualize the filters only for the nodes that are involved in the decision-making 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 analysed by using post-hoc techniques such as Grad-CAM (31). In this example, a saliency map is generated for the concept "Broken pedimen". Today, this doorway can be visited at the "Capilla Real" in Granada, Spain. (4).

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.

7.4.7. 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 Figure 7.6 we present a prototype of a user interface inspired by (35). As it can be observed in the 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

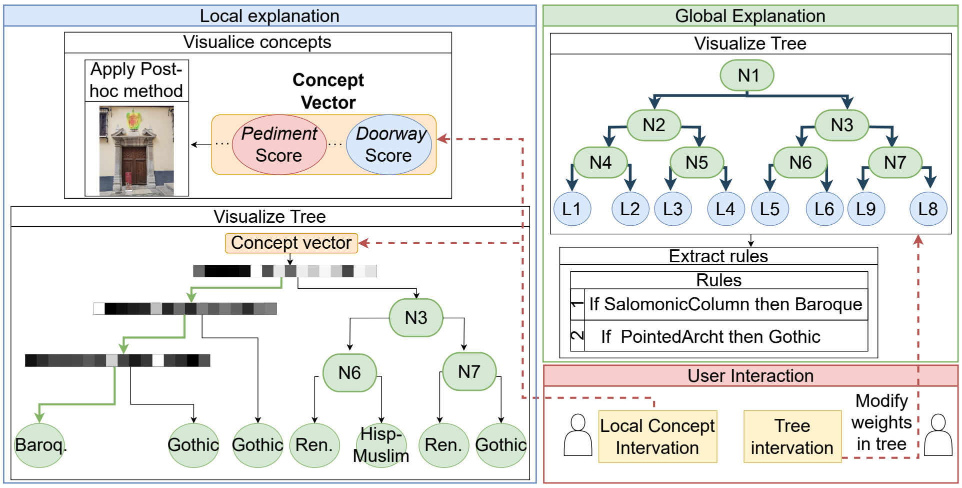

Figure 7.6: User interface prototype: the user interface should offer functionality for three main tasks: global explanations, local explanations and user intervention.

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 visualise 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.

7.5. 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 (21, 27, 13). 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.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[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. *Information Fusion*, 58:82–115, 2020.

[3] Adrien Bennetot, Gianni Franchi, Javier Del Ser, Raja Chatila, and Natalia Díaz-Rodríguez. Greybox xai: a neural-symbolic learning framework to produce interpretable predictions for image classification. *Knowledge-Based Systems*, 258:109947, 2022.

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

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. URL `https://arxiv.org/abs/1802.02611`.

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and 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*, pages 1971–1978, 2014.

[7] Natalia Díaz-Rodríguez, Alberto Lamas, Jules Sanchez, Gianni Franchi, Ivan Donadello, Siham Tabik, David Filliat, Policarpo Cruz, Rosana Montes, and Francisco Herrera. Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. *Information Fusion*, 79:58–83, 2022.

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

[9] Ivan Donadello and Luciano Serafini. Integration of numeric and symbolic information for semantic image interpretation. *Intelligenza Artificiale*, 10(1):33–47, 2016.

[10] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

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

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

[13] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations, 2019.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[15] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explai-

ning black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

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

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

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, pages 1–15, 2015.

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

[20] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*, pages 1467–1475, 2015.

[21] Ashish Kumar, Karan Sehgal, Prerna Garg, Vidhya Kamakshi, and Narayanan C Krishnan. Mace: Model agnostic concept extractor for explaining image classification networks, 2020.

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

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[25] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[26] Christoph Molnar. *Interpretable machine learning.* Lulu. com, 2020.

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

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

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

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

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

[32] Timo Speith. A review of taxonomies of explainable artificial intelligence (xai) methods. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2239–2250, 2022.

[33] Pradyumna Tambwekar, Andrew Silva, Nakul Gopalan, and Matthew Gombolay. Natural language specification of reinforcement learning policies through differentiable decision trees. *IEEE Robotics and Automation Letters*, 2023.

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

[35] AMM Sharif Ullah and Khalifa H Harib. A human-assisted knowledge extraction method for machining operations. *Advanced Engineering Informatics*, 20(4):335–350, 2006.

[36] Warren J von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

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

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

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

7.6. Statistical tests

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 7.4: Paired t-test Joint-Tree compared to Joint-Baseline (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 7.5: Paired t-test Joint-Tree compared to Sequential-Tree (MonuMAI)

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 7.6: Paired t-test Joint-Tree compared to Joint-Baseline (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 7.7: Paired t-test Joint-Tree compared to Ind-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	

Table 7.8: Paired t-test Joint-Tree compared to Sequential-Tree (PASCAL).

Chapter 8

Concept logic trees

David Morales[1,2], M.P. Cuellar[2], Diego P. Morales[3].

1. HAT.tec, Lilienthalstraße 15, 85579 Neubiberg, Germany
2. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
3. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

Applied Intelligence

- Accepted: 4 February 2024 Published 21 February 2024
- DOI: 10.1007/s10489-024-05321-4
- Impact factor: 5.3
- JCR Rank: 48/145 in category COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE (Q2)

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.

8.1. 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, 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 9.2. Next, the proposed approach is presented in Section 9.3. Then, in Section 9.4 we describe the experiments and discuss the results. Finally, Section 9.5 presents future research directions and conclusions.

8.2. 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 Koh et al. (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 (6, 3).

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, Koh et al. (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 (35, 34, 2). 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 Badreddine et al. (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

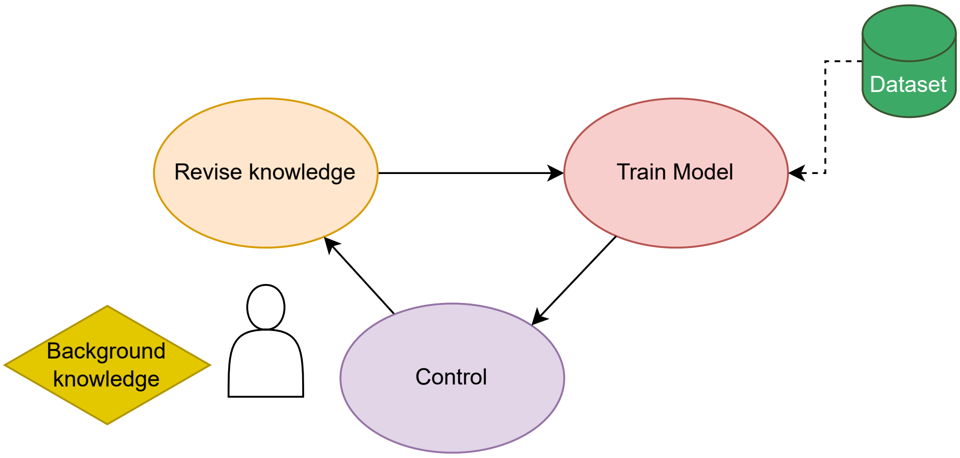

Figure 8.1: Learning cycle. The use of neural symbolic learning enables human intervention. An expert can control the training process and intervene the model by revising the knowledge through the definition or re-definition of knowledge rules.

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 Figure 8.1.

8.3. Methodology

In this section we present our proposed solution that corresponds to the learning cycle presented in Figure 8.1. 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 8.2 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.

8.3.1. 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 $t(g(\mathbf{x}))$, where $g : \mathbb{R}^d \to [0,1]^k$ maps the image from the input space to the concept space. A classification subnetwork $t : [0,1]^k \to \mathbb{Y}$ 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 $L_Y : \mathbb{Y} \times \mathbb{Y} \to \mathbb{R}^+$, 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 $L_c : [0,1]^k \times [0,1]^k \to \mathbb{R}^+$ 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.

8.3.2. 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:

$$p_i(\mathbf{x}) = \theta(\mathbf{x}\mathbf{w}_i + \mathbf{b}_i) \tag{8.1}$$

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:

$$Q^l = \mathrm{softmax}(\phi^l) \tag{8.2}$$

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

$$L_T(\mathbf{x}) = -\sum_{l \in L} P^l(\mathbf{x}) \sum_{k \in Y} y_k \log Q^l_k \tag{8.3}$$

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

$$P^l(\mathbf{x}) = \prod_N p_i(\mathbf{x})^{\mathbf{1}[l \swarrow i]} (1 - p_i(\mathbf{x}))^{\mathbf{1}[i \searrow l]} \tag{8.4}$$

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).

8.3.3. 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

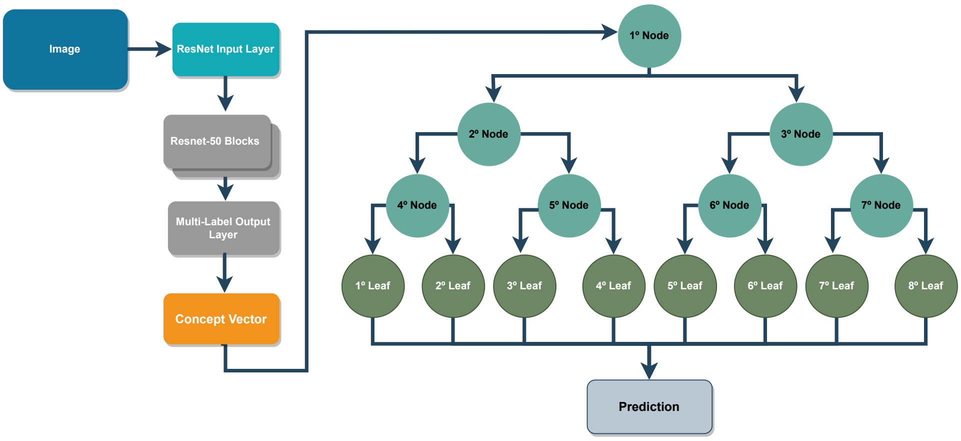

Figure 8.2: Architecture diagram of the proposed solution. The concept extractor g (implemented by a Resnet-50) takes an image as input and outputs the concept vector. The binary soft decision tree t gets the concept vector as input and outputs the final prediction.

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:

$$L_{Sym} = 1 - Sat_K(x, \phi) \tag{8.5}$$

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 Figure 8.2). 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.

8.3.4. 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 Figure 8.2) according to the learning cycle introduced in Figure 8.1. For this purpose, we combine the loss functions presented before in the subsections 8.3.1, 8.3.2 and 8.3.3. The equation for the proposed approach is presented in 8.6:

$$\sum_{n=1}^{N} (\alpha L_Y(t(g(\mathbf{x}_n)); y_n) + \beta L_{Sym}(\mathbf{x}_n, \phi) + \delta L_c(g(\mathbf{x}_n); \mathbf{c}_n)) \tag{8.6}$$

where the hyperparameters $\alpha, \beta, \delta > 0$ control the trade-off between the losses. In Figure 8.3 a diagram of the training process is presented. The

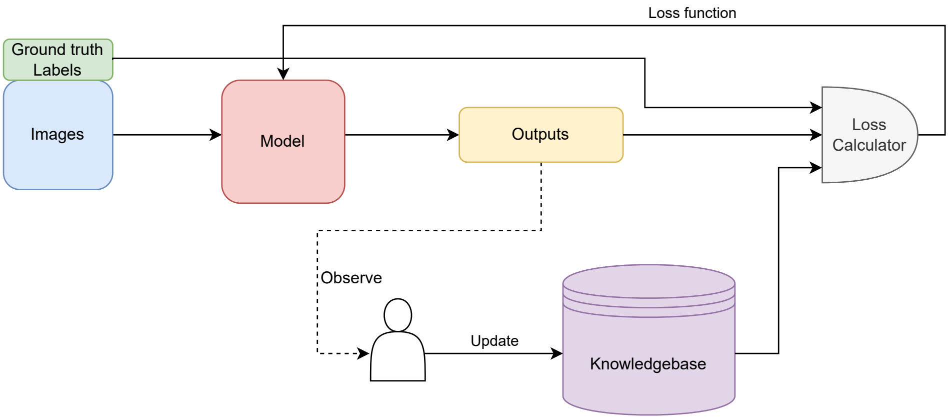

Figure 8.3: Diagram of the proposed training process. The user can redefine the knowledge base by adding or modifying rules. These rules are used to compute the symbolic loss L_{Sym} as explained in Section 8.3.3. The final loss function is calculated aggregating all losses L_Y, L_c and L_{Sym} as explained in equation 8.6.

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 equation 8.6. 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 Figure 8.1. Furthermore, we explore the combination of neural symbolic learning with each of the three possible concept-bottleneck models described in Section 8.3.1. 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 8.3.1. The classifier training would be done by minimizing the equation 8.7:

$$\sum_{n=1}^{N}(\alpha L_Y(t(\mathbf{c}_n); y_n) + \beta L_{Sym}(\mathbf{x}_n, \phi)) \tag{8.7}$$

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 8.3.1.

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

8.4.1. 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 (6, 3).

8.4.2. 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).

8.4.3. Predicates and rules

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

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

- 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" (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 Table 8.4 and Table 8.5. This knowledge is extracted from the descriptions provided by the authors of the datasets (21, 8) 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	$SegmentalArch$	$Baroque \vee Renaissance$

Figure 8.4: Knowledge base for the MonuMAI dataset. This knowledge base contains all knowledge rules for the MonuMAI dataset. See Subsection 8.4.3.

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	$Screen$	$Tvmonitor$

Figure 8.5: Knowledge base for the PASCAL dataset. This knowledge base contains all knowledge rules for the PASCAL dataset. See Subsection 8.4.3.

8.4.4. Experiments and Results

In this section, we introduce the experiments carried out to validate the proposed methods (see section 9.3) on the datasets presented in Section 9.4.1. We kept the splits in training and test sets that were proposed in (3, 6).

8.4.5. 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.

8.4.5.1. 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 8.1. 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.

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

Table 8.1: 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. Best results for each evaluation measurement are in bold.

8.4.5.2. 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 approaches (satisfiability for the knowledge rules and classical approach for the classes). The results are presented in Table 8.2. Please note that these results correspond to models where no knowledge in form of attribute rules is present.

	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**

Table 8.2: 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. Best results for each evaluation measurement are in bold.

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.

8.4.6. Results

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

In Table 9.1 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 9.3, 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

$$\text{Y-Acc} = Acc(y, y') \tag{8.8}$$

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

$$\text{C-Acc} = Acc(\mathbf{c}, \mathbf{c}') \tag{8.9}$$

where *boldc* 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.

$$Sat = Sat_K(\mathbf{x}, \phi) \tag{8.10}$$

We repeated every experiment three times and present the mean results in Table 9.1. 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 8.3.4 and 9.4. 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 8.3.1).

MonuMAI

	Base-Ind	Ind-N	Base-Seq	Seq-N	Base-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

	Base-Ind	Ind-N	Base-Seq	Seq-N	Base-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**

Table 8.3: Results of the proposed experiments on both datasets for the already presented metrics. Best results for each evaluation measurement are in bold.

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 same since the same concept extractor g is used for both models, and only t is different. Please refer to Section 9.3 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 8.5. 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 9.3). 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 Subsection 8.3.1) 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).

8.4.7. 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, 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 8.4.3 (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 8.3.1) (18). We compute a separate satisfiability "Sat Path" for the new rules. We present the results in Table 8.4.

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

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

Table 8.4: Results of the proposed experiments on both datasets for the already presented metrics. Best results for each evaluation measurement are in bold.

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.

8.4.8. 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 9.2. Three of the models are transparent models (Greybox (3), EXPLANet (6)) and (28) and the other two models are opaque models (DeiT-B (33, 3) 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 8.5. 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 (28, 33, 3). 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

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 (33, 3)	96,48	**90,85**
MonuNet (21)	83,11	-

Table 8.5: 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.

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 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.

8.4.9. 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 Miller (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 8.4.7. Additionally, our model is compatible with the user concept intervention as shown in (18).

8.5. 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 (20, 30, 13). 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.

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability and access

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

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.

Ethical and informed consent for data used

Not applicable.

References

[1] 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. *Information Fusion*, 58:82–115, 2020.

[2] Samy Badreddine, Artur dÁvila Garcez, Luciano Serafini, and Michael Spranger. Logic tensor networks. *Artificial Intelligence*, 303:103649, 2022.

[3] Adrien Bennetot, Gianni Franchi, Javier Del Ser, Raja Chatila, and Natalia Díaz-Rodríguez. Greybox xai: a neural-symbolic learning framework to produce interpretable predictions for image classification. *Knowledge-Based Systems*, 258:109947, 2022.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 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*, pages 833–851. Springer, 2018.

[5] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and 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*, pages 1971–1978, 2014.

[6] Natalia Díaz-Rodríguez, Alberto Lamas, Jules Sanchez, Gianni Franchi, Ivan Donadello, Siham Tabik, David Filliat, Policarpo Cruz, Rosana Montes, and Francisco Herrera. Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: The monumai cultural heritage use case. *Information Fusion*, 79:58–83, 2022.

[7] Zihan Ding. Popular-rl-algorithms. `https://github.com/quantumiracle/Popular-RL-Algorithms`, 2019.

[8] Ivan Donadello and Luciano Serafini. Integration of numeric and symbolic information for semantic image interpretation. *Intelligenza Artificiale*, 10(1):33–47, 2016.

[9] Ivan Donadello, Luciano Serafini, and Artur DÁvila Garcez. Logic tensor networks for semantic image interpretation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1596–1602, 2017.

[10] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

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

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

[13] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

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

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

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, pages 1–15, 2015.

[18] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.

[19] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulo. Deep neural decision forests. In *Proceedings of the IEEE international conference on computer vision*, pages 1467–1475, 2015.

[20] Ashish Kumar, Karan Sehgal, Prerna Garg, Vidhya Kamakshi, and Narayanan Chatapuram Krishnan. Mace: Model agnostic concept extractor for explaining image classification networks. *IEEE Transactions on Artificial Intelligence*, 2(6):574–583, 2021.

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

[22] Fabrizio Lamberti, Lia Morra, and Filomeno Davide Miro. End-to-end training of logic tensor networks for object detection. 2021.

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[25] Simone Martone, Francesco Manigrasso, Fabrizio Lamberti, and Lia Morra. Prototypical logic tensor networks (proto-ltn) for zero shot learning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4427–4433. IEEE, 2022.

[26] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[27] Christoph Molnar. *Interpretable machine learning.* Lulu. com, 2020.

[28] David Morales Rodríguez, Manuel Pegalajar Cuellar, and Diego P Morales. On the fusion of soft-decision-trees and concept-based models. *Available at SSRN 4402768.*

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

[30] Andres Felipe Posada-Moreno, Nikita Surya, and Sebastian Trimpe. Eclad: Extracting concepts with local aggregated descriptors, 2023.

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

[32] Timo Speith. A review of taxonomies of explainable artificial intelligence (xai) methods. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 2239–2250, 2022.

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers &amp; distillation through attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/touvron21a.html`.

[34] Benedikt Wagner and AS dÁvila Garcez. Neural-symbolic integration for fairness in ai. In *CEUR Workshop Proceedings*, volume 2846, 2021.

[35] Benedikt Wagner and Artur dÁvila Garcez. Neural-Symbolic Integration for Interactive Learning and Conceptual Grounding, January 2022. URL `http://arxiv.org/abs/2112.11805`. arXiv:2112.11805 [cs].

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

[37] Shunli Wang, Yongcun Fan, Siyu Jin, Paul Takyi-Aninakwa, and Carlos Fernandez. Improved anti-noise adaptive long short-term memory neural network modeling for the robust remaining useful life prediction of lithium-ion batteries. *Reliability Engineering & System Safety*, 230: 108920, 2023.

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

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

Chapter 9

Exploring methods for the generation of visual counterfactuals in the latent space.

David Morales[1,2], M.P. Cuellar[2], Diego P. Morales[3].

1. HAT.tec, Lilienthalstraße 15, 85579 Neubiberg, Germany
2. Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
3. Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

This article is under review.

Abstract:

In the field of eXplainable Artificial Intelligence (XAI), the generation of counterfactuals is a promising method for human-interpretable explanations. A counterfactual explanation describes a causal situation in the form: “If X had not occurred, Y would not have occurred”. In this work, we study the generation of visual counterfactuals in the latent space for deep learning image classification models. We explore how to adapt the training environment to facilitate the generation of counterfactuals, combining ideas coming from different fields such as multitasking or generative learning, with the aim of developing more interpretable models. We study well-known counterfactual methods and how to apply them in the latent space. Furthermore, we propose a new way of generating counterfactuals working in the latent space and compare it with the other studied approaches, achieving competitive results.

9.1. 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* machine learning models, such as Artificial Neural Networks (ANNs). To achieve a higher degree of interpretability, new techniques and models have been proposed in the last few years with the aim of developing more interpretable artificial intelligence (3). Most of these techniques provide heat maps or saliency maps to identify the regions of the input images that ANNs look at when making predictions. However these saliency maps do not answer the question “Why class A and not class B?” (22). Counterfactuals aim to explain the behaviour of machine learning methods by generating changes to the input that would cause a different behavior of the system (4). Given an input data point and a model, a counterfactual is defined as a generated data point that is as close to the input data point as possible but for which the model gives a different outcome (15). The generation of counterfactuals is usually used as a post-hoc explanation technique that allows the user to inspect the model in order to understand its behavior. This way, the user of a black box model could benefit of these techniques by using them interpretate and understand the model’s decision process. Most of these techniques work in an iterative loop where they perturb the input image an feed the classifier with the new image until the classifier changes its decision. This may drive to situations where noise is added to the original image until the classifier changes its decision, similar as done in adversarial learning, resulting in adversarial samples and not in explanations (6). This issue is likely to happen if the counterfactual technique does not really understand the mechanisms of the classifier and how it internally works. But this is difficult as the post hoc counterfactual generation technique is applied after the classifier is defined and trained, being usually seen as a black box by the counterfactual technique. This problem would be solved if the classifier were interpretable, as no post hoc methods would be needed. The definition of interpretable and transparent models and methods that achieve state-of-the-art performance is one of the main goals of XAI, but it is still a distant goal in the field of deep learning. In the meanwhile, we introduce a solution that aims to be a step in that direction. We propose to train image classifiers in an environment that prepare them for the generation of counterfactuals. The proposed environment allow the counterfactual techniques to work in the latent space without iteratively directly perturbing the image, what could lead to adversarial samples. We are motivated by two points that were already pointed out: firstly, most of the known explanation techniques are post-hoc methods (5). We believe that machine learning models should be designed and trained to be interpretable. Secondly, many counterfactual generation techniques change the classifier decision by perturbing the input image in an iterative loop, what may result in adding noise in a similar way as in adversarial learning (6), producing poor explanations. In this work, we define a new metric that measures the answer of the classifier when the counterfactuals are denoised. Furthermore, we explore how modifying known counterfactual generation techniques to apply them in the latent space may lead to avoid this issue. More specifically, we propose to modify two of these well known techniques for this purpose and to adapt the training scheme to prepare the image classifier for the generation of visual counterfactuals in the latent space. Furthermore, we propose a new method for resolving this task. We compare the proposed solution with other well-known approaches and achieve competitive results. The main contributions of this research work are:

1. A new architecture and training approach in the field of image classification to prepare the models for the generation of bluevisual counterfactuals in the latent space, generating interpretable models.

2. The proposal of a novel metric to discover counterfactuals generated in an adversarial setting.

The remaining of the manuscript is organized as follows. Section 9.2 includes an overview of the state of the art. Section 9.3 presents the proposed training approach. Section 9.4 introduces the two image datasets, describes the experiments carried out, and analyzes the obtained results. Finally, Section 9.5 closes with the conclusions and future lines of research.

9.2. Related Work

While the very first machine learning systems were easily interpretable, the last few years have been characterized by an upsurge of opaque decision systems, such as deep neural networks (DNNs) (3, 5). DNNs are the state-of-the-art on many machine learning tasks due to their great generalization and prediction skills. However, they are considered *black-box* machine learning models. In this context, there has been a growing influx of work on explainable artificial intelligence. In the field of image classification, post-hoc explanations, which refer to the use of interpretation methods after training a model, and feature relevance methods are increasingly the most adopted approaches to explain DNNs (3). Most of these explanation techniques provide heat maps to identify the regions of the input images that networks look at when making predictions, allowing the data to be interpreted at a glance. Note that these heat maps are also referred to in the literature as sensitivity maps, saliency maps, or class activation maps. Some well-known visual explanation techniques are Class Activation Mapping (CAM) (24) or Local Interpretable Model-agnostic Explanations (LIME) (19). These saliency maps techniques aim to answer the question of -What is the model looking at? or -What features are relevant to the model?. However, the questions of -Why was this image classified as “A” and not as “B”? or -What has to change in the image to be classified as “B”? remains open (22). The use of visual counterfactuals is a promising explainability approach that has been studied in recent years for black-box models and aims to answer these questions. Wachter et al. (21) proposed to define a loss function where a first term guides the search towards points which would change the prediction and a second term ensures that the counterfactual is close to the original instance. Counterfactuals were generated by using ADAM to optimize the loss function. Later, Looveren and Klaise (14) proposed the use of class prototypes to guide the search of counterfactuals. These prototypes can be generated by training an autoencoder on the same dataset and using the mean encoding of the instances which belong to that class. Once the prototypes are generated, they defined an optimization problem, similar to (21), adding a prototype loss term. The prototype loss term forces the counterfactual to be near to the target prototype. Apart of using generative learning for generating prototypes, other authors have explored generative AI models for the generation of counterfactuals in the field of image classification. In this area, we highlight the work done by Singla et al. (20) who proposed to use a generative adversarial network GAN to explain a binary classifier. They produced a progressive set of perturbations to the query image that gradually changes the posterior probability from its original class to its negation. On the other hand, counterfactuals and visual explanation methods constitute post-hoc explanation techniques, but the problem of defining self-explaining deep learning models is still open. A first solution was proposed by Alvarez Melis and Jaakkola

(2), who presented 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 by using an autoencoder and train 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 problematic of this approach relies on how understable and appropriate these concepts are. The concepts could be defined by an expert, but this would require data annotations and human intervention.

In this article, we investigate how to train an image classifier and produce counterfactuals in a self-explaining way. Inspired by (20) and (2), we investigate the use of a conditional generator to generate visual counterfactuals by modifying the latent representations, making use of a multitasking environment and an autoencoder architecture, to train the model with both tasks in mind: classification and visual counterfactual explanation. While in the field of image classification most explainability techniques (including counterfactuals methods) focus on post-hoc explanation, our method is a step in the direction of building more interpretable architectures, where the classifier is trained together with the counterfactual model and both share the same latent space.

9.3. Methodology

In this section, we describe the proposed approach for training an interpretable classifier that is able to generate counterfactuals. We also describe different approaches to generate counterfactuals in the latent space.

9.3.1. Training environment

Our aim is to define and train a classifier that is prepared to generate counterfactuals to explain its behavior. The idea is to train the classifier and at the same time, face two tasks: 1) prepare the latent space for the generation of counterfactuals and 2) train a decoder that transfers vectors from the latent space to the image space. This decoder will be later used to explore counterfactuals in the latent space and transfer them to the original image space. To better understand this idea, it is useful to know that it is inspired and based on variational auto-encoders and how they are used in generative learning (11, 18, 8).

We train the classifier together with the decoder in a multitask environment. As already introduced, we based our architecture on variational autoencoders and use the Kullback Leibler divergence (KLd) as regularization term to ensure that the latent space follows a Gaussian distribution, as usually done in generative learning. The classifier structure consists of an encoder and

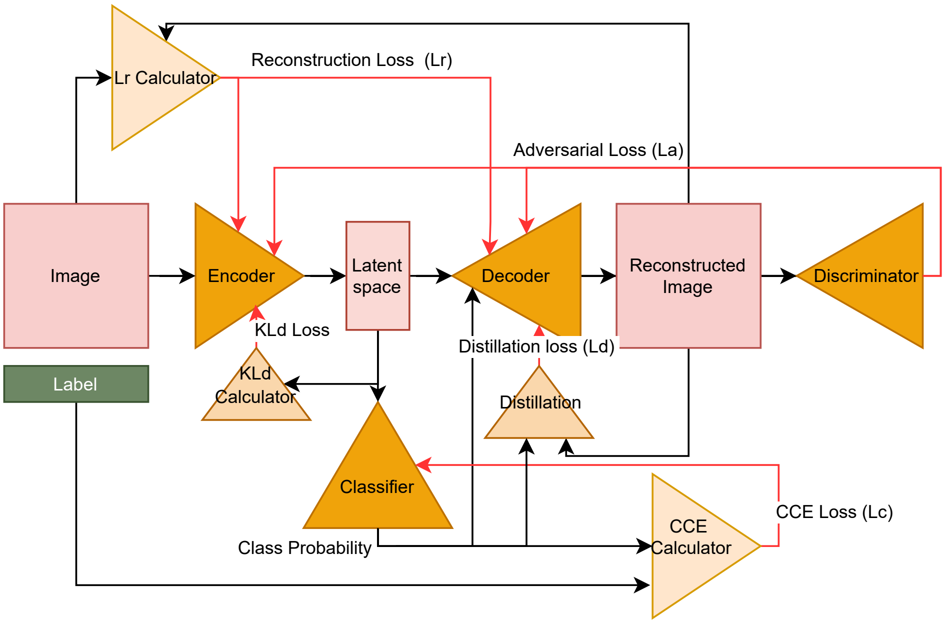

Figure 9.1: Architecture of the proposed approach. The black arrows represent the information flow, while the red arrows represent the computed losses (see equations 9.1, 9.2 and 9.3). Loss calculators, modules (i.e. Encoder) and data (i.e. Image) are shown in different colors.

a classification sub-network as usual. The space where the encoder has its output and the classifier its input is the same, and we will refer to it as the latent space. We link the decoder to the encoder. This means that the decoder has also its input in the latent space, resulting in a latent space shared by the three modules. We train the entire structure facing two tasks: the classification task and a reconstruction task. The encoder provides the classifier and the decoder with a feature map. Then the classifier uses this feature map to infer the class, while the decoder aims to reconstruct the original image. If we keep this decoder setup, it would be similar to the usual setup when using VAEs in generative learning. But we want it to explain the classifier's output. This is why we condition the decoder on the classifier's output, similar to (20).

Once we have explained the architecture, now we describe the loss functions that should be used during training. We first describe the loss function used to train the decoder (L_d) and that is proposed in equation 9.1. It is known that making use of pixel-wise metrics in reconstruction tasks may lead to blurry reconstructions (17). That is why the decoder is trained in an adversarial environment, by making use of a discriminator to compute an adversarial loss (L_a) that should force the decoder to generate realistic images. This is, the decoder gets feedback from the discriminator, which is trained to discriminate real images from fake images. This is the usual approach when using GANs (7). To ensure that the generated images are not only realistic but also similar to the original image, we propose using the classifier to compute a loss $KLd(y', y'_r)$ that helps us to reconstruct the image, in addition to a pixel-wise loss function (L_r). Inspired by the distillation algorithms, we computed this loss applying the Kullback Leibler divergence on the output (y') of the classifier on the original image and on the output (y'_r) of the classifier on the generated image. Note that y_r is the result of feeding the classifier with the reconstructed image. This loss also has the objective of forcing the decoder and the classifier to understand the latent space in the same way. For the classification task, the loss used L_c is the categorical cross-entropy (CCE) as usual (see 9.2). To train the encoder, we use the loss function L_e presented in equation 9.3. The first term of the equation $KLd(z_m, z_v)$ is the Kullback Leibler divergence that ensures that the latent space follows a Gaussian distribution and is widely used when training variational autoencoders (10). The second term corresponds to the classifier's loss, that is, the categorical crossentropy. The second and the third terms are the adversarial loss and the reconstruction loss, that are also present in equation 9.1 and that were already explained. In figure 9.1 we provide a diagram where we present the explained architecture and losses and describe how the information flows.

$$L_d = L_a(x, x') + L_r(x, x') + KLd(y', y'_r) \tag{9.1}$$

9.1: Decoder loss (L_d): the decoder's training loss is computed by aggregating the adversarial loss, the reconstruction loss and the kullback Leibler divergence.

$$L_c = CCE(y, y') \tag{9.2}$$

9.2: Classifier loss (L_c): we use the categorical crossentropy, which is a usual way of computing classification loss.

$$L_e = KLd(z_m, z_v) + L_c(y, y') + L_a(x, x') + L_r(x, x') \tag{9.3}$$

9.3: Encoder loss (L_d): we combine the KLd, the classifier loss, the adversarial loss and the reconstruction loss.

9.3.2. Counterfactuals generation

Once we have trained the model, the decoder and the classifier use the same latent space, which is based on VAEs as explained before. Thanks to the Kullback-Leibler divergence that we used as the regularization term, the latent space follows a Gaussian distribution. This allows us to use the decoder in a similar way as done with VAEs in generative learning. Based on that, we propose to modify the latent vectors and feed the decoder with them to produce counterfactuals. We explore different alternatives:

1. Use of Cf-Wachter: Make use of the algorithm proposed in (21) in the latent space in order to generate new vectors in that space. To this aim, we adapt the code provided in Alibi [1].

2. Use of prototypes: Use the algorithm proposed in (14). The algorithm is applied to the vectors in the latent space. To this aim, we adapt the code provided by the authors in Alibi [2].

3. Iterative: We propose this new approach which is inspired on iterated functions. In the following we define the latent prototype $proto^i$ of a class i as the point in the latent space that correspond to the mean of all encoded instances of that class. That is

$$proto^i = \sum_{img \in X^i} \frac{E(img)}{n_i} \tag{9.4}$$

[1] https://docs.seldon.io/projects/alibi/en/latest/methods/CF.html

[2] https://docs.seldon.io/projects/alibi/en/latest/methods/CFProto.html

where X^i are the images in the training set that belong to the class i and E is the encoder. Our hypothesis is that for each class, the latent prototype should encode features of the class that are common to most of the instances of that class. We can see this point as an attractor point, as the decoder D may tend to reconstruct images similar to the one encoded by this point. Actually what we are saying is that rare features are more difficult to be learned than features that appear frequently. Our approach is based on the idea of seeing and using the autoencoder $A = D \circ E$ as an iterated function $Counterfactual = (A \circ A \circ ... \circ A)(img)$, aiming to follow the way out from the original class to the target class. As the decoder is conditioned on the class probability (see Figure 9.1), we can modify this probability in order to increase it for the target class and to decrease it for the original class. Being P_t the probability of the target class and P_o the probability of the original class, we update the probabilities $P_t = \alpha * P_o + (1-\alpha) * P_t$ and $P_t = \alpha * P_t + (1-\alpha) * P_o$, where $\alpha \in [0,1]$ is a parameter that decreases its value in each iteration in order to increase the probability P_t. To generate a counterfactual for an original image, we fed the decoder with this new probability and with the original latent vector. In a first iteration, the algorithm may not be able to converge to the target class. That is why we recursively repeat the process by feeding the model with the generated images until it converged or until a maximum of iterations is reached. Algorithm 2 shows the pseudo-code of the proposed method.

9.4. 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 two experiments carried out, including the evaluation metrics considered. Finally, we report and analyze the results obtained in both experiments.

9.4.1. Datasets

We evaluated our proposed method on two well-known datasets: the MNIST dataset (13), and the Fashion MNIST dataset (23). The MNIST is a dataset of handwritten digits while the Fashion-MNIST is a dataset of Zalando's article images. Both datasets have a similar structure: a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. We normalize the images in the range of 0 to 1.

```
Data: Image, encoder, decoder, classifier, desiredClassProbability,
      maxIter
Result: Counterfactual
encoded_vector = encoder(Image);
clfPrediction = classifier(encoded_vector);
for iter in range(0,maxIter) do
    counterfactual = decoder([encoded_vector,
     desiredClassProbability]);
    encoded_vector = encoder(counterfactual);
    prediction_counterfactual = classifier(encoded_vector);
    if prediction_counterfactual is desired output then
        return counterfactual;
    else
        iter=iter+1;
        update desiredClassProbability;
    end
end
```

Algorithm 2: Pseudo-code of the proposed method to generate counterfactuals.

9.4.2. Implementation details

Regarding the network architecture, the encoder has 3 convolutional layers and one final dense layer. The decoder has a mirror structure, with upsampling2D layers after the convolutional layers. The classification subnetwork is composed of two dense layers and a softmax activation output. The discriminator network has two convolutional layers, two dense layers, and one softmax output. The dimension of the latent space is 16 for the MNIST dataset and 20 for the Fashion MNIST. We used the Adam optimization algorithm (9) with learning rate 10^{-3} for all networks except for the discriminator. For the discriminator the learning rate was set to 10^{-5} as the discriminator task is easier than the generation task. Regarding the training step, we used a batch size of 124. The training process finished after 100 epochs. The proposed method was implemented on TensorFlow and Keras.

9.4.3. Experiments

This section describes the experiment designed to evaluate the proposed methods (see section 9.3). We tested the proposed approach on both datasets individually. Furthermore, we employed the methods Cf-Wachter and Prototypes such as described in the original articles to generate counterfactuals to compare them to the proposed approach. Furthermore, we adapted these two methods to apply them in the latent space. We call them Latent-Cf-Wachter (L-Cf-Wachter) and Latent-Prototypes (L-Prototypes). Our implementation of those two methods is based on the code provided in (12). The use of the Cf-Wachter method in the latent space is straightforward, we only need to adapt the inputs and outputs dimensions to the latent space. Then we use the encoder to get the encoded vectors in the latent space and feed the algorithm with these vector and after that we generate counterfactuals by applying the decoder on the outputs. For the L-Prototypes algorithm, we modify the code to get the prototypes in the latent space for each class and after that, apply the algorithm in this space, using the encoder and the decoder as done for L-Cf-Wachter. We generate counterfactuals using these two methods and compare them to the already mentioned approaches. We kept the original split in training and test sets for the two considered datasets (see Section 9.4.1). Then, we trained the model using the training set. We evaluate counterfactual generation methods on the isolated test set, using the performance metrics described in Section 9.4.4. When generating the counterfactuals, we choose as target class the nearest class. In the case of the two algorithms using prototypes, this class is defined as the class whose prototype is the nearest one to the sample and different from the original class. For the other algorithms, this class is chosen as the class different from the original class, for what the classifier predicts a higher score.

9.4.4. Evaluation

To evaluate the performance of the proposed models and make a fair comparison with other approaches, we computed some popular metrics for the task of counterfactual generation. Furthermore, we define a new metric inspired by the tendency of some counterfactual techniques to add noise in an adversarial way to change the classifier decision.

- Realism: a measure of how well a counterfactual "fits in" with the known data distribution (16). A denoising autoencoder AE $(\cdot)$ is trained on the training set. The L2 norm of the reconstruction error is used as a measure of realism. A lower value represents higher realism.

$$Realism = \mathbf{E}[||AE(x_i^{cf}) - x_i^{cf}||_2^2] \tag{9.5}$$

- Actionability (Act): a measure of the distance between the counterfactual (x_i^{cf}) and the input data point (x_i) using the L1 norm (16).

$$Actionability = \mathbf{E}[||x_i^{cf} - x_i||_1] \tag{9.6}$$

- Failure rate (FR): the percentage of times that the counterfactual technique employed does not achieve to change the classifier decision. This means that the generated counterfactual is classified in the original class.

$$FR = \frac{\sum g(C(x_i), C(x_i^{cf}))}{n} \quad g(p,q) = \begin{cases} 1 & if \quad p = q \\ \\ 0 & if \quad p \neq q \end{cases} \tag{9.7}$$

- Denoised Failure Rate (DFR): we define this metric that aims to show if a counterfactual was generated in an "adversarial way" by adding noise to change the classifier decision. Similar as for realism, we use a denoising AE and calculate the failure rate over the denoised counterfactuals. We use the same AE as used for realism.

$$DFR = FR(AE(X)) \tag{9.8}$$

- Time: time in seconds needed to generate counterfactuals

9.4.5. Results

In this section, we report and analyze the results obtained in the experiments described in Section 9.4.3.

In Table 9.1 we present the results obtained for the different methods. It can be observed that the method Cf-Wachter obtained the best results for actionability. This was expected as this metric aims to measure the L1 distance between the generated counterfactual and the original image. This means that the counterfactuals generated by this method are very close (pixel-wise) to the original images. However, this method is the one with the poorest results for the DFR metric, going from 0 FR to 1 DFR (remember this metric is the result of getting the FR for the denoised counterfactuals). These two facts make us think that the method is generating counterfactuals by adding "noise" to the original image. When we modify this method to apply it over the latent space (L-Cf-Wachter), the results obtained for realism and for DFR improve considerably. We can observe that in this case, the FR is almost equal to the denoised FR (DFR). However, the actionability is increased and the FR is higher, meaning that this method does not always achieve to generate a counterfactual that actually changes the classifier decision. The results obtained for the metric realism show us that the counterfactuals generated by the L-Cf-Wachter method are closer to the original distribution than the ones generated by Cf-Wachter.

When we compare the results obtained for Prototypes and for L-prototypes (Prototypes applied latent spaces), the behaviour is similar to the one observed for Cf-Wachter and L-Cf-Wachter. When applying the method in the latent space, the results obtained for realism and DFR are better than when applying the original method, but the actionability increases. This method improves the results obtained by the L-Cf-Wachter for realism, while getting worse results for FR and DFR. The actionability is similar for both

methods. Looking at the results obtained for Prototypes, we can observe that this method obtains better DFR than Cf-Wachter.
The proposed approach gets better FR and Actionability than the other "latent" methods while getting better results for realism and DFR than the not latent methods. Comparing it to the methods that better perform for each metric, this method achieves comparable or better results for the metricts realism, FR and DFR.

Mnist

	Proto	L-Proto	Cf-Wacht.	L-Cf-Wacht.	Iterative
Realism	1,29	**0,69**	0,98	0,77	0.76
Act.	4,28	8,42	**1,38**	8,24	7.67
FR	**0 %**	32,07 %	**0 %**	14,68 %	**0 %**
DFR	77,40 %	32,77 %	97,77 %	**16,2 %**	18,6 %
Time	10,09	4,05	7,08	5,08	**0,47**

Fashion

	Proto	L-Proto	Cf-Wacht.	L-Cf-Wacht.	Iterative
Realism	0,84	**0,60**	1,04	0,66	0.64
Act.	4,27	10,65	**1,78**	11,53	10.50
FR	**0 %**	36,85 %	0,02 %	9,16 %	1.58 %
DFR	70,07 %	36,87 %	74,17 %	9,11 %	**8,8 %**
Time	10,22	4,06	7,65	4,75	**0,59**

Table 9.1: Results of the proposed experiments on both datasets for the already presented metrics. Best results are in bold.

Regarding to the time required to generate counterfactuals with each of the studied methods, the results show that the mean time required when working on the latent space is in general lower than when generating the counterfactuals in the original space. In this scenario, the proposed method is faster than the baseline approaches.

9.4.6. Visual analysis

Figure 9.2 shows counterfactuals generated using the five proposed methods. They were picked up from the counterfactuals generated for the experiments described in Section 9.4.4. It can be seen that the counterfactuals generated by using the Wachter method (b) contain a lot of noise, which is consistent with the results presented in Table 9.1 and with the analysis done before: this counterfactual technique tends to generate adversarial samples. These adversarial samples achieve to fool the classifier by adding noise to the original image. The noise added to the image is barely noticeable for humans, who can clearly advice that there is no significant change in the image. We believe that these kind of counterfactuals do not help the user to unders-

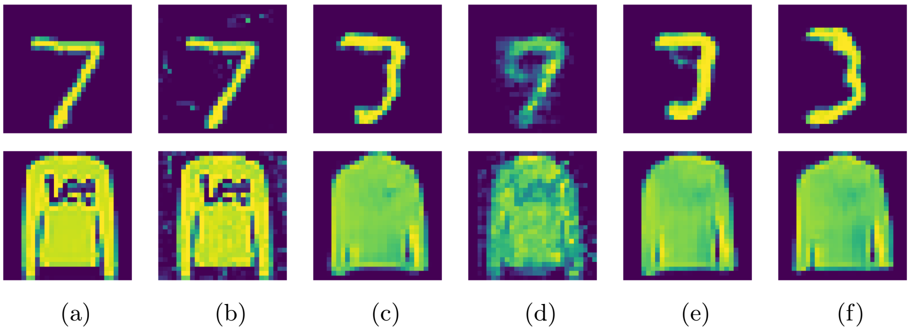

Figure 9.2: (a) Input images from the MNIST (top) and Fashion-MNIST (bottom) datasets, counterfactuals generated for the input images with the methods: (b) Wachter , (c) L-Wachter, (d) Prototypes, (e) L-Prototypes, Iterative (f).

tand what should be different in the image in order to change the decision of the classifier. Neither to inspect the model to understand what features are important for the decision. When applying this counterfactual techniques in the latent space (c), the counterfactuals are not generated by adding noise anymore. These counterfactuals help to understand how the image could be modified to be classified as a digit "7" in the case of the MNIST dataset or as a "coat" in the case of the Fashion dataset. These counterfactuals could help the user to inspect the model in order to understand its behavior.

The Prototypes method (d) generates counterfactuals that also contain a lot of noise. They cannot be clearly classified as adversarial samples as they modify the original image and incorporate features that allow to understand the behavior of the classifier. However the noise does not allow us to clearly identify what should be different in the image in order to change the decision of the classifier. Applying this method in the latent space (e) leads to generate better understandable counterfactual, similar to the ones generated by L-Wachter (c).

The counterfactuals generated using the iterative method (f) are similar to the counterfactuals generated by the other two latent methods. One different that can be observed in the case of the MNIST sample is that the counterfactuals generated by this method are closer to the original distribution, what is consistent with the results obtained for the metric Actionability in Section 9.4.5.

9.5. Conclusions and future work

In this research work, we explore how to define and train an interpretable classification convolutional model by preparing it for the generation of counterfactuals. This is achieved by defining an architecture based on VAEs and generative learning and by training the model in an multitasking approach. This all results in an architecture where the latent space is shared by the encoder, the classifier and the decoder. This allows us to explore the latent space and to generate counterfactual in that space. Our proposed aims to be a step in the direction of self-explaining methods and makes of our approach not a pure post-hoc approach. Furthermore, the decision of generating counterfactuals in the latent space is motivated by the known issue that some counterfactual generation techniques may change the classifier decision by adding noise similar as in adversarial learning. In order to study this issue, we define a new metric that makes use of a denoising autoencoder. We explore how modifying these techniques to apply them in the latent space may lead to avoid this issue. More specifically, we propose to modify two of these well known techniques to generate counterfactuals in the latent space and show how this can help to alleviate this issue. Furthermore, we propose a new way of generating counterfactuals in the latent space. We compare the proposed solution with the already mentioned methods and achieve competitive results. Furthermore, the proposed technique shows to be notably faster than the other studied approaches.

In future work, we will continue exploring the latent space of other models such as generative models. We think that models can benefit from having a rich latent space in order to make them more interpretable and self-explaining. We believe that knowledge can be transferred between tasks, by sharing layers or the latent space or by training the models in multitasking environment. We would like continue exploring this research area in order to explore more interpretable architectures and models. All the models used in this work were defined ad hoc. The use and combination of already defined architectures for classification and generation would allow us to face more complex environments and data. We would like to focus our future research on style GANs, where the latent space is known to be really rich and disentangled. Additionally, the use of prototypes in datasets with a high intra-class variation is a challenging task. One prototype per class can not represent a high intra-class variation (1). Our proposed solution shares this issue as a prototype based solution. In future work we would like to study possible extensions of our proposal in order to face this challenge.

Conflict of interest

The authors declare that they have no conflict of interest.

Data

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

References

[1] Saeid Abbaasi, Kamaledin Ghiasi-Shirazi, and Ahad Harati. A multi-prototype capsule network for image recognition with high intra-class variations. *Neural Processing Letters*, pages 1–15, 2023.

[2] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

[3] 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. *Information Fusion*, 58:82–115, 2020.

[4] André Artelt, Valerie Vaquet, Riza Velioglu, Fabian Hinder, Johannes Brinkrolf, Malte Schilling, and Barbara Hammer. Evaluating robustness of counterfactual explanations. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–09. IEEE, 2021.

[5] Adrien Bennetot, Jean-Luc Laurent, Raja Chatila, and Natalia Díaz-Rodríguez. Towards explainable neural-symbolic visual reasoning. *IJCAI Neural-Symbolic Learning and Reasoning Workshop*, 2019.

[6] Timo Freiesleben. The intriguing relation between counterfactual explanations and adversarial examples. *Minds and Machines*, pages 1–33, 2021.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. ISSN 0001-0782. doi: 10.1145/3422622. URL `https://doi.org/10.1145/3422622`.

[8] Salman H. Khan, Munawar Hayat, and Nick Barnes. Adversarial training of variational auto-encoders for high fidelity image generation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1312–1320, 2018. doi: 10.1109/WACV.2018.00148.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations*, pages 1–15, 2015.

[10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes in 2nd international conference on learning representations. In *ICLR 2014-Conference Track Proceedings*, 2014.

[11] Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

[12] Janis Klaise, Arnaud Van Looveren, Giovanni Vacanti, and Alexandru Coca. Alibi explain: Algorithms for explaining machine learning models. *Journal of Machine Learning Research*, 22(181):1–7, 2021. URL `http://jmlr.org/papers/v22/21-0017.html`.

[13] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[14] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer, 2021.

[15] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[16] Daniel Nemirovsky, Nicolas Thiebaut, Ye Xu, and Abhishek Gupta. Countergan: Generating realistic counterfactuals with residual generative adversarial nets. *arXiv preprint arXiv:2009.05199*, 2020.

[17] Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

[18] Mehran Pesteie, Purang Abolmaesumi, and Robert N. Rohling. Adaptive augmentation of medical data using independently conditional variational auto-encoders. *IEEE Transactions on Medical Imaging*, 38(12): 2807–2820, 2019. doi: 10.1109/TMI.2019.2914656.

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

[20] Sumedha Singla, Brian Pollack, Stephen Wallace, and Kayhan Batmanghelich. Explaining the black-box smoothly-a counterfactual approach. *arXiv preprint arXiv:2101.04230*, 2021.

[21] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[22] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[23] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

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

Compadre, quiero cambiar
mi caballo por su casa,
mi montura por su espejo,
mi cuchillo por su manta.
Compadre, vengo sangrando,
desde los montes de Cabra.
Si yo pudiera, mocito,
ese trato se cerraba.
Pero yo ya no soy yo,
ni mi casa es ya mi casa.

Romance Sonámbulo
Federico García Lorca