Neural Prototype Trees for Interpretable Fine-grained Image Recognition

Meike NAUTA [a,b,1], Ron VAN BREE [a] and Christin SEIFERT [b]
[a] *University of Twente, Enschede, The Netherlands*
[b] *University of Duisburg-Essen, Essen, Germany*

Abstract. Interpretable machine learning addresses the black-box nature of deep neural networks. Visual prototypes have been suggested for intrinsically interpretable image recognition, as alternative to post-hoc explanations that only approximate a trained model. Aiming for better interpretability and fewer prototypes to not overwhelm a user, we propose the Neural Prototype Tree (ProtoTree), a deep learning method that includes prototypes in a hierarchical decision tree to faithfully visualize the entire model. In addition to global interpretability, a path in the tree explains a single prediction. Each node in our binary tree contains a trainable prototypical part. The presence or absence of this learned prototype in an image determines the routing through a node. Decision making is therefore similar to human reasoning: Does the bird have a red throat? And an elongated beak? Then it's a hummingbird! We tune the accuracy-interpretability trade-off using ensembling and pruning. We apply pruning without sacrificing accuracy, resulting in a small tree with only 8 learned prototypes along a path to classify a bird from 200 species. An ensemble of 5 ProtoTrees achieves competitive accuracy on the CUB-200-2011 and Stanford Cars data sets. Code is available at `https://github.com/M-Nauta/ProtoTree`. Full paper published at CVPR 2021.

Keywords. interpretable machine learning, explainable AI, fine-grained image recognition, decision tree, prototypes

1. Introduction

There is an ongoing scientific dispute between simple, interpretable models and complex black boxes, such as Deep Neural Networks (DNNs). DNNs have achieved superior performance, but their complexity has led to an increasing demand for interpretability [1]. In contrast, decision trees are easy to understand and interpret [2,3], because they transparently arrange decision rules in a hierarchical structure. Their predictive performance is however far from competitive for computer vision tasks. We address this so-called 'accuracy-interpretability trade-off' [1,4] by combining the expressiveness of deep learning with the interpretability of decision trees.

We present the *Neural Prototype Tree*, ProtoTree in short, an intrinsically interpretable method for fine-grained image recognition. A ProtoTree has the representational power of a neural network, and contains a built-in binary decision tree structure, as shown

[1] Corresponding author: Meike Nauta, E-mail: m.nauta@utwente.nl

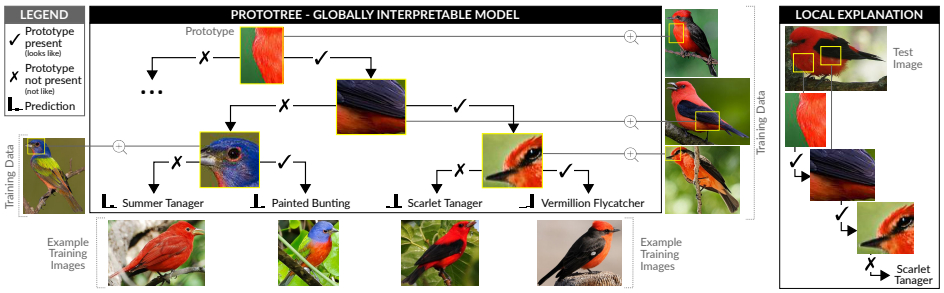

Figure 1. Example of a ProtoTree. A ProtoTree is a globally interpretable model faithfully explaining its entire behaviour (left, partially shown) and additionally the reasoning process for a single prediction can be followed (right): the presence of a red chest and black wing, and the absence of a black stripe near the eye, identifies a Scarlet Tanager.

in Fig. 1 (left). Each internal node in the tree contains a trainable *prototype*. Our prototypes are prototypical *parts* learned with backpropagation, as introduced in the Prototypical Part Network (ProtoPNet) [5] where a prototype is a trainable tensor that can be visualized as a patch of a training sample. The extent to which this prototype is present in an input image determines the routing of the image through the corresponding node. Leaves of the ProtoTree learn class distributions. The paths from root to leaves represent the learned classification rules. To this end, a ProtoTree consists of a Convolutional Neural Network (CNN) followed by a binary tree structure and can be trained end-to-end with a standard cross-entropy loss function. We only require class labels and do not need any other annotations. To make the tree differentiable and back-propagation compatible, we utilize a *soft* decision tree, meaning that a sample is routed through both children, each with a certain weight. We present a novel routing procedure based on the similarity between the latent image embedding and a prototype.

A ProtoTree approximates the accuracy of non-interpretable classifiers, while being *interpretable-by-design* and offering truthful global and local explanations. This way it provides a novel take on interpretable machine learning. In contrast to *post-hoc* explanations, which approximate a trained model or its output [6,7], a ProtoTree is inherently interpretable since it directly incorporates interpretability in the structure of the predictive model. A ProtoTree therefore faithfully shows its entire classification behaviour, independent of its input, providing a *global* explanation (Fig. 1). As a consequence, our compact tree enables a human to convey, or even print out, the *whole* model. In contrast to *local* explanations, which explain a single prediction and can be unstable and contradicting [8,9], global explanations enable *simulatability* [4]. Additionally, our ProtoTree can produce *local* explanations by showing the routing of a specific input image through the tree (Fig. 1, right). Hence, ProtoTree allows retraceable decisions in a human-comprehensible number of steps. In case of a misclassification, the responsible node can be identified by tracking down the series of decisions, which eases error analysis.

Scientific Contributions

- We present an intrinsically interpretable neural prototype tree architecture for fine-grained image recognition.
- Outperforming ProtoPNet [5] while having roughly only 10% of the number of prototypes, included in a built-in hierarchical structure.

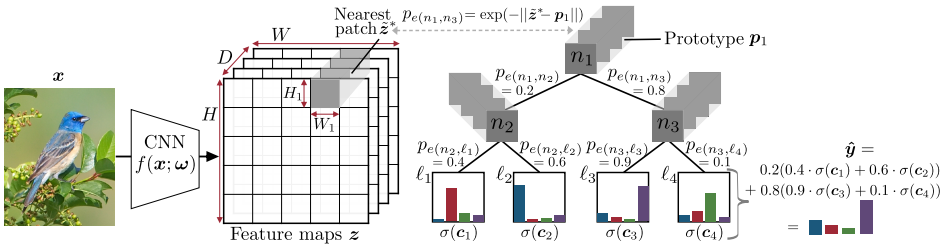

Figure 2. Decision making process of a ProtoTree to predict class probability distribution $\hat{\boldsymbol{y}}$ of input image $\boldsymbol{x}$. During training, prototypes $\boldsymbol{p}_n$, leaves' class distributions $\boldsymbol{c}$ and CNN parameters $\boldsymbol{\omega}$ are learned. Probabilities p_e (shown with example values) depend on the similarity between a patch in the latent input image and a prototype.

- An ensemble of 5 interpretable ProtoTrees achieves competitive performance on CUB-200-2011 [10] (CUB) and Stanford Cars [11].

2. Neural Prototype Tree

A ProtoTree combines a Convolutional Neural Network (CNN) with a binary tree. In contrast to traditional decision trees, where a node routes a sample either right or left, our ProtoTree is *soft* during training and routes a sample to both children, each with a certain probability that together sum to 1. As shown in Figure 2, the CNN outputs a set of feature maps, $\boldsymbol{z}$. Each node n in the tree contains a prototype $\boldsymbol{p}_n$, which is a trainable tensor that is visualised after training. The depths of $\boldsymbol{p}_n$ and $\boldsymbol{z}$ are identical, such that the distance can be calculated between a patch in $\boldsymbol{z}$ and $\boldsymbol{p}_n$. This distance is converted to a similarity score within $[0, 1]$ that determines to what extent the sample is routed through the right edge. Multiplying all probabilities along a path results in the probability that $\boldsymbol{z}$ ends up in a leaf. Subsequently, a weighted combination of all leaf distributions results in the final prediction, as visualized in Figure 2.

Our CNN and all prototypes are jointly optimized with backpropagation. Class distributions in the leaves are learnt with a derivative-free algorithm. After training, the learned prototypes are visualized by upsampling them to a patch from the nearest training image. We reduce tree size by pruning ineffective prototypes. Furthermore, the soft ProtoTree can be converted to a hard, and therefore more interpretable, tree without loss of accuracy. Lastly, to increase predictive power, we can create an ensemble by averaging the predictions of multiple ProtoTrees.

3. Experiments and Results

We compare our ProtoTree with ProtoPNet [5] (an interpretable model which uses a bag of class-specific prototypical parts) and a state-of-the-art black box. We evaluate on CUB-200-2011 [10] with 200 bird species and Stanford Cars [11] with 196 car types. As ProtoTree's CNN backbone, we use a pre-trained ResNet50 architecture.

Table 1 shows that our ProtoTree outperforms ProtoPNet [5] in both accuracy and interpretability. Whereas ProtoPNet presents a user an overwhelming number of proto-

Data set	Method	Interpret-ability	Top-1 Acc.	#Prototypes
CUB (224×224)	Triplet Model [12]	-	**87.5**	n.a.
	ProtoPNet [5]	+	79.2	2000
	ProtoTree (ours)	++	82.2±0.7	**202**
	ProtoPNet ensemble (3) [5]	+	84.8	6000
	ProtoTree ensemble (3)	+	86.6	605
	ProtoTree ensemble (5)	+	**87.2**	1008
CARS (224×224)	RAU [13]	-	**93.8**	n.a.
	ProtoPNet [5]	+	86.1	1960
	ProtoTree (ours)	++	86.6±0.2	**195**
	ProtoPNet ensemble (3) [5]	+	91.4	5880
	ProtoTree ensemble (3)	+	90.3	586
	ProtoTree ensemble (5)	+	**91.5**	977

Table 1. Our ProtoTree and ensemble with 3 or 5 ProtoTrees compared with uninterpretable state-of-the-art (-) and interpretable prototype-based models (+, ++).

types, we improve interpretability by arranging the prototypes in a hierarchical tree structure. This breaks up the reasoning process in small steps which simplifies model comprehension and error analysis. After pruning the tree, the number of prototypes is a factor of 10 smaller than ProtoPNet. An ensemble of 5 ProtoTrees approximates the accuracy of uninterpretable state-of-the-art, while still having fewer prototypes than ProtoPNet.

Deterministic reasoning. A ProtoTree can be converted from a soft to a hard tree to make deterministic predictions at test time. Selecting the leaf with the highest path probability leads to nearly the same accuracy, since the fidelity (i.e., fraction of test images for which the soft and hard strategy make the same classification [3]) is 0.999 for a ProtoTree of height 9 trained on CUB. A greedy strategy performs slightly worse, but still has a fidelity of 0.987. Results are similar for other datasets and tree heights, showing that a ProtoTree can be safely converted to a deterministic tree, such that a prediction can be explained by presenting one path in the tree. A deterministic ProtoTree (height = 9), reduces the number of decisions to follow to 9 prototypes at maximum. When using a more accurate ensemble of 5 deterministic ProtoTrees, a maximum of only 45 prototypes needs to be analysed, resulting in much smaller local explanations than ProtoPNet.

4. Conclusion

We presented the Neural Prototype Tree (ProtoTree) for intrinsically interpretable fine-grained image recognition. our novel architecture with end-to-end training procedure improves interpretability by arranging the prototypes in a hierarchical tree structure. This breaks up the reasoning process in small steps, thereby enhancing understandability and reducing local explanation size. Our ProtoTree achieves competitive performance while maintaining intrinsic interpretability. As a result, our work questions the existence of an accuracy-interpretability trade-off and stimulates novel usage of powerful neural networks as backbone for interpretable, predictive models.

References

[1] Adadi A, Berrada M. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access. 2018;6.

[2] Freitas AA. Comprehensible Classification Models: A Position Paper. SIGKDD Explor Newsl. 2014 Mar;15(1):1–10. Available from: https://doi-org.ezproxy2.utwente.nl/10.1145/2594473.2594475.

[3] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM computing surveys (CSUR). 2018;51(5):1-42.

[4] Lipton ZC. The Mythos of Model Interpretability. Queue. 2018 Jun;16(3):30:31-0:57.

[5] Chen C, Li O, Tao D, Barnett A, Rudin C, Su JK. This Looks Like That: Deep Learning for Interpretable Image Recognition. In: Advances in Neural Information Processing Systems 32; 2019. Available from: http://papers.nips.cc/paper/9095-this-looks-like-that-deep-learning-for-interpretable-image-recognition.pdf.

[6] Montavon G, Samek W, Müller KR. Methods for interpreting and understanding deep neural networks. Digital Signal Processing. 2018;73:1-15.

[7] Laugel T, Lesot MJ, Marsala C, Renard X, Detyniecki M. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. arXiv preprint arXiv:190709294. 2019.

[8] Alvarez-Melis D, Jaakkola TS. On the robustness of interpretability methods. arXiv preprint arXiv:180608049. 2018.

[9] Kindermans PJ, Hooker S, Adebayo J, Alber M, Schütt KT, Dähne S, et al. In: Samek W, Montavon G, Vedaldi A, Hansen LK, Müller KR, editors. The (Un)reliability of Saliency Methods. Cham: Springer International Publishing; 2019. p. 267-80. Available from: https://doi.org/10.1007/978-3-030-28954-6_14.

[10] Wah C, Branson S, Welinder P, Perona P, Belongie S. The Caltech-UCSD Birds-200-2011 Dataset. California Institute of Technology; 2011. CNS-TR-2011-001.

[11] Krause J, Stark M, Deng J, Fei-Fei L. 3D Object Representations for Fine-Grained Categorization. In: 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13). Sydney, Australia; 2013. .

[12] Liang J, Guo J, Guo Y, Lao S. Adaptive Triplet Model for Fine-Grained Visual Categorization. IEEE Access. 2018;6.

[13] Ma X, Boukerche A. An AI-based Visual Attention Model for Vehicle Make and Model Recognition. In: 2020 IEEE Symposium on Computers and Communications (ISCC); 2020. p. 1-6.