PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification

Meike Nauta[1], Jörg Schlötterer[2,3], Maurice van Keulen[1], Christin Seifert[2,3]
[1]University of Twente, the Netherlands
[2]University of Duisburg-Essen, Germany, [3]University of Marburg, Germany
m.nauta@utwente.nl, joerg.schloetterer@uni-marburg.de,
m.vankeulen@utwente.nl, christin.seifert@uni-marburg.de

Abstract

Interpretable methods based on prototypical patches explain their reasoning by recognizing various components in an image. Existing part-prototype methods can learn prototypes that are not in line with human visual perception, making interpretation not intuitive. Driven by the principle of explainability-by-design, we introduce PIP-Net (Patch-based Intuitive Prototypes Network): an interpretable image classifier that learns part-prototypes in a self-supervised fashion which correlate better with human vision. PIP-Net can be interpreted as a sparse scoring sheet where the presence of a prototypical part (concept) adds evidence for a class. PIP-Net can also abstain from a decision for out-of-distribution data by saying "I haven't seen this before". We only use image-level labels and do not rely on any part annotations. The learned prototypes show the entire reasoning of the model. A smaller local explanation locates the relevant prototypes in one image. We show that our prototypes correlate with ground-truth object parts. Accepted to CVPR 2023. Code is available at `https://github.com/M-Nauta/PIPNet`.

1. Introduction

There is high demand for understanding the reasoning of deep neural networks [9]. In contrast to common post-hoc explainability that reverse-engineers a black box, we take interpretability as a design starting point for in-model explainability. Driven by the recognition-by-components theory [1], we introduce PIP-Net: Patch-based Intuitive Prototypes Network. It automatically identifies semantically meaningful components, while only having access to image-level class labels and not relying on part annotations. The components are "prototypical parts" (prototypes) visualized as image patches, which separate the decision process in multiple steps (see Fig. 1). PIP-Net is globally interpretable and designed to be highly intuitive with its simple scoring-sheet reasoning: the presence of relevant prototypical parts

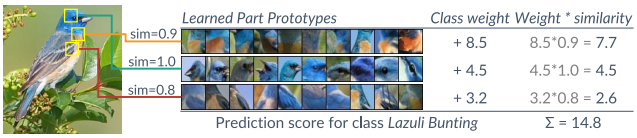

Figure 1. Compact local explanation of PIP-Net. PIP-Net learns part-prototypes and localizes them in an unseen input image.

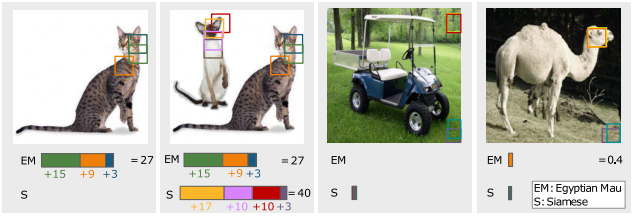

Figure 2. PIP-Net is an intuitive single-object classifier that can handle multi-object images and out-of-distribution data. It can abstain from a decision and instead say "I haven't seen this before".

increases class evidence. When no relevant prototypes are present in the image, with *e.g.* out-of-distribution (OoD) data, PIP-Net will abstain from a decision (Fig. 2).

Recent interpretable part-prototype models are ProtoPNet [2], ProtoTree [10] and ProtoPool [12]. They are designed for fine-grained image recognition (birds and car types) and regularize interpretability on class-level by assuming that (parts of) images from the same class have the same prototypes. This assumption may however not hold, leading to a lack of "semantic correspondence" [7] between learned prototypes and human concepts [4, 5] (see also Fig. 3). PIP-Net addresses this "semantic gap" gap between latent and pixel space. It is designed to be intuitive and optimized to correlate with human vision. A sparse linear layer connects learned interpretable prototypical parts to classes. A user only needs to inspect the prototypes and their relation to the classes in order to interpret the model. The linear layer can be interpreted as a scoring sheet: the score for a class is the sum of all present prototypes multiplied by

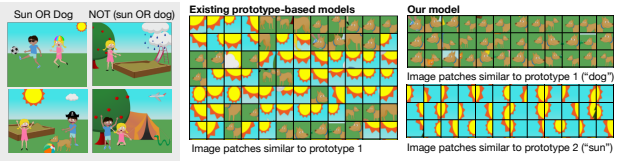

Figure 3. Toy dataset [14] with two classes (left). Existing models can learn representations of prototypes that do not align with human visually perceived similarity (center). PIP-Net learns prototypes that represent concepts that also look similar to humans (right).

their weights. Due to our design, PIP-Net can detect that an image does not belong to *any* class (OoD) or that it belongs to *multiple* classes. The scoring sheet is trained with novel regularization that specifically optimizes for interpretability. Instead of specifying the number of prototypes beforehand as in ProtoPNet [2], PIP-Net selects as few prototypes as possible for good classification accuracy with compact explanations, reaching sparsity ratios $> 99\%$. Because of our interpretable *and* predictive linear layer, we ensure a direct relation between explanation and classification.

2. PIP-Net: Training and Reasoning

PIP-Net has a CNN backbone (we use ConvNeXt-tiny [8] with adapted strides) that learns an interpretable, 1-dimensional image encoding $\boldsymbol{p}$ indicating the presence or absence of prototypical parts. For example, $\boldsymbol{p}$ could be $[0.9, 0.0, 0.1, 0.8]$, indicating that the first and fourth prototype are present in this image. A sparse linear layer with non-negative weights connects prototypical parts to classes.

Training of PIP-Net consists of two stages. First, self-supervised contrastive learning optimizes the prototypes for semantic similarity, independent of the classification task. We apply different data augmentations to an input image to create a positive pair. We optimize PIP-Net to assign the same prototype to two views of an augmented image patch, thereby incorporating human perception into the training process. In the second stage, we add a negative log-likelihood loss that trains the linear layer and finetunes the pre-trained prototypes to be relevant for the classification task.

For intuitive and compact scoring-sheet reasoning, PIP-Net is trained with a novel custom activation function. Naively applying softmax would allow to train with the regular negative log-likelihood loss, but conflicts with our goals of compactness and decision abstaining. An overconfident softmax score would give no incentive to further reduce weights for other classes. Prototypes which are actually irrelevant for a class might therefore keep a positive weight, which results in explanations that are larger than necessary. Simply normalizing before softmax would impede interpretability since prototype scores with a value of zero become non-zero, thereby losing the OoD detection

	Method	Accuracy	#Global	#Local
CUB	PIP-Net	84.3±0.2	495±6	10 (4)
	ProtoPNet [2]	79.2	2000	2000
	ProtoTree [10]	82.2±0.7	202	8.3
	ProtoPool [12]	85.5±0.1	202	202
CARS	PIP-Net	88.2±0.5	515±4	9 (4)
	ProtoPNet [2]	86.1	1960	1960
	ProtoTree [10]	86.6±0.2	195	8.5
	ProtoPool [12]	88.9±0.1	195	195

Table 1. Mean accuracy across 3 random seeds (↑). Global size (↓): total number of relevant prototypes in the model. Local size (↓): number of non-zero prototypes for a single prediction: for all classes in total, and between brackets for the predicted class only.

property. To prevent unexpected and unintuitive behavior, prototype presence scores should stay zero and behave independently of each other. Hence, output scores are calculated as $\boldsymbol{o} = \log((\boldsymbol{p}\boldsymbol{\omega}_c)^2 + 1)$, where $\boldsymbol{p}$ are the prototype presence scores and $\boldsymbol{\omega}_c$ the weights of the linear layer. The natural logarithm reduces large weights, such that the model is incentivized to reduce the weights of irrelevant prototypes. This novel normalization step therefore implicitly optimizes for smaller explanations. During inference, the output scores are simply calculated as $\boldsymbol{p}\boldsymbol{\omega}_c$ for easy interpretation.

3. Experiments, Results and Conclusion

We evaluate our model on CUB-200-2011 [13] (200 bird species), Stanford Cars [6] (196 car models), Oxford-IIIT Pet [11] (37 cat and dog species) and PartImageNet [3]. PIP-Net achieves competitive classification accuracy, while having a low number of prototypes (see Tab. 1). We quantify the OoD-detection with the common FPR@95-metric by determining class-specific thresholds for output score $\boldsymbol{o}$ such that 95% of the ID samples are classified as in-distribution. We find that PIP-Net gives zero-scores to 77%-99% of the OoD samples. To quantify the semantic correspondence between prototypes and image patches, we measure the *purity* of prototypes with ground-truth annotations of object parts available in the CUB and PartImageNet datasets. We evaluate to what extent the top-10 image patches for a prototype (as shown in Fig. 1) represent the same part. We find that the purity of PIP-Net, up to 93%, is substantially higher than purities of existing part-prototype models. This result aligns with visual analysis of the prototypes and confirms the interpretability of the learned prototypes.

We think that interpretability-by-design should become the new standard for interpretable/explainable AI. This approach resulted in PIP-Net, which provides compact explanations that align with human perception, allowing to interpret decisions in an intuitive, faithful and meaningful way.

References

[1] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 1

[2] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1, 2

[3] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pages 128–145. Springer, 2022. 2

[4] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021. 1

[5] Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, Ruth Fong, and Olga Russakovsky. HIVE: Evaluating the human interpretability of visual explanations. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[6] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013. 2

[7] Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions, 2022. 1

[8] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, June 2022. 2

[9] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Comput. Surv.*, feb 2023. 1

[10] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14933–14943, June 2021. 1, 2

[11] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. 2

[12] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 351–368, Cham, 2022. Springer Nature Switzerland. 1, 2

[13] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2

[14] C. L. Zitnick and Devi Parikh. Bringing semantics into focus using visual abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 2