Interpretable Deep Learning with Prototypical Part

Bartosz Zieliński

gmum group of machine learning research

About me

Education

- Habilitation (2022)
    - Wrocław University of Science and Technology
    - **Explainable and interpretable machine learning with biomedical applications**
- Ph.D. (2012)
    - Institute of Fundamental Technological Research, Polish Academy of Science
    - **Detection of selected rheumatoid lesions based on hand radiographs**
- MSc (2007)
    - Faculty of Mathematics and Computer Science, Jagiellonian University
    - **Automatic detection of joint space narrowing in metacarpophalangeal and interphalangeal joints based on hand radiographs**

Occupation

- Associate Professor
  - Faculty of Mathematics and Computer Science, Jagiellonian University
- Research Team Leader
  - IDEAS NCBR sp. z o. o.
- Computer Vision Expert
  - Ardigen S.A.

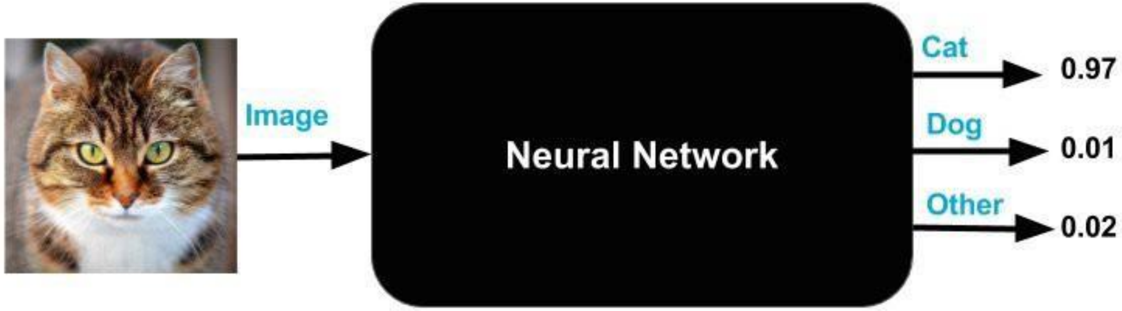
Research interests

- eXplainable Artificial Intelligence (XAI)
- Sustainable machine learning
- Effective deep learning training
- Training deep learning in demanding setups

Motivation

Motivation

- Deep learning is widely used due to its superior performance
- However, it suffers from the lack of interpretability (caused by the black-box character of standard deep neural networks)



Wrong decisions can be costly and dangerous

Missouri S&T News and Research

After Uber, Tesla incidents, can artificial intelligence be trusted?

Apr 10, 2018

BBC NEWS

Tay: Microsoft issues apology over racist chatbot fiasco

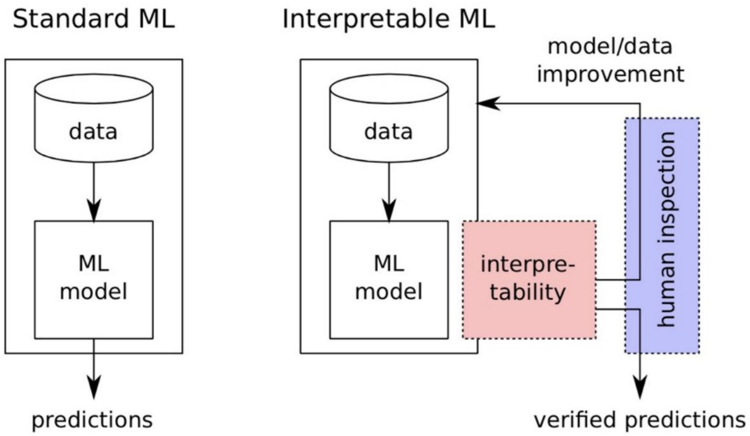
Sep 22, 2017

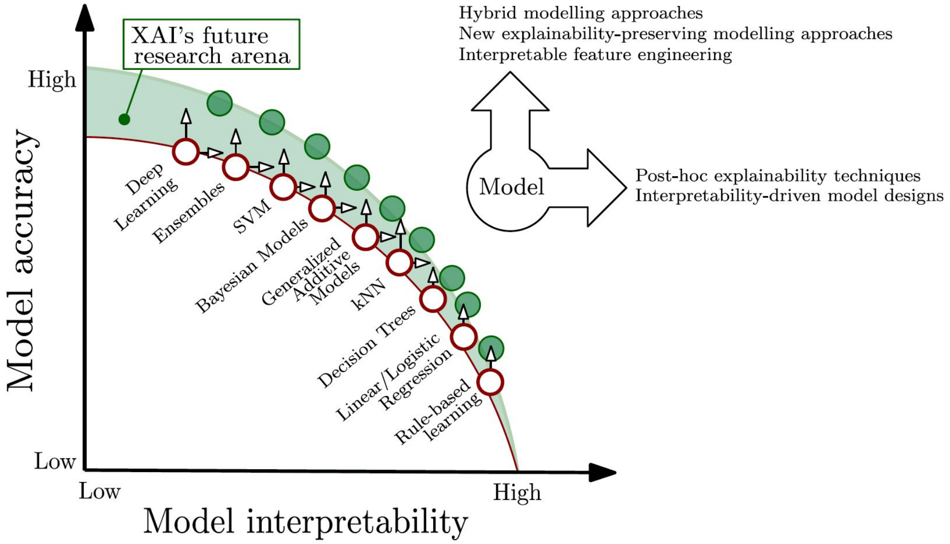
Guilty! AI Is Found to Perpetuate Biases in Jailing

1 day ago

https://xaitutorial2020.github.io/raw/master/slides/aaai_2020_xai_tutorial.pdf

Explainable AI (post-hoc vs. self-explainable)



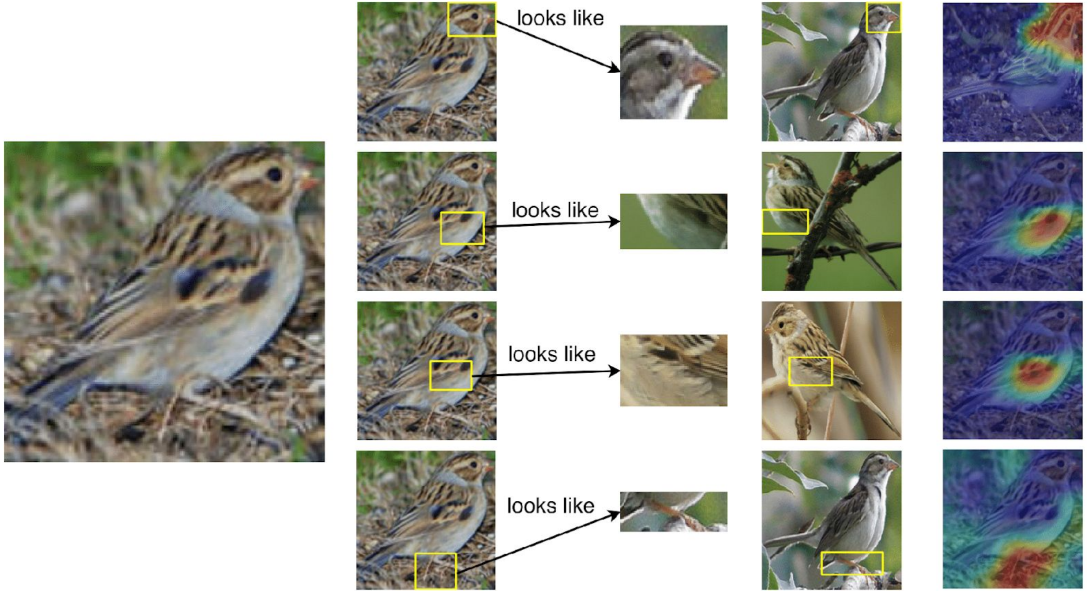


https://xaitutorial2020.github.io/raw/master/slides/aaai_2020_xai_tutorial.pdf

Arrieta et al. Explainable artificial intelligence: Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion, 2020

Prototypical Parts Network (ProtoPNet)

Idea



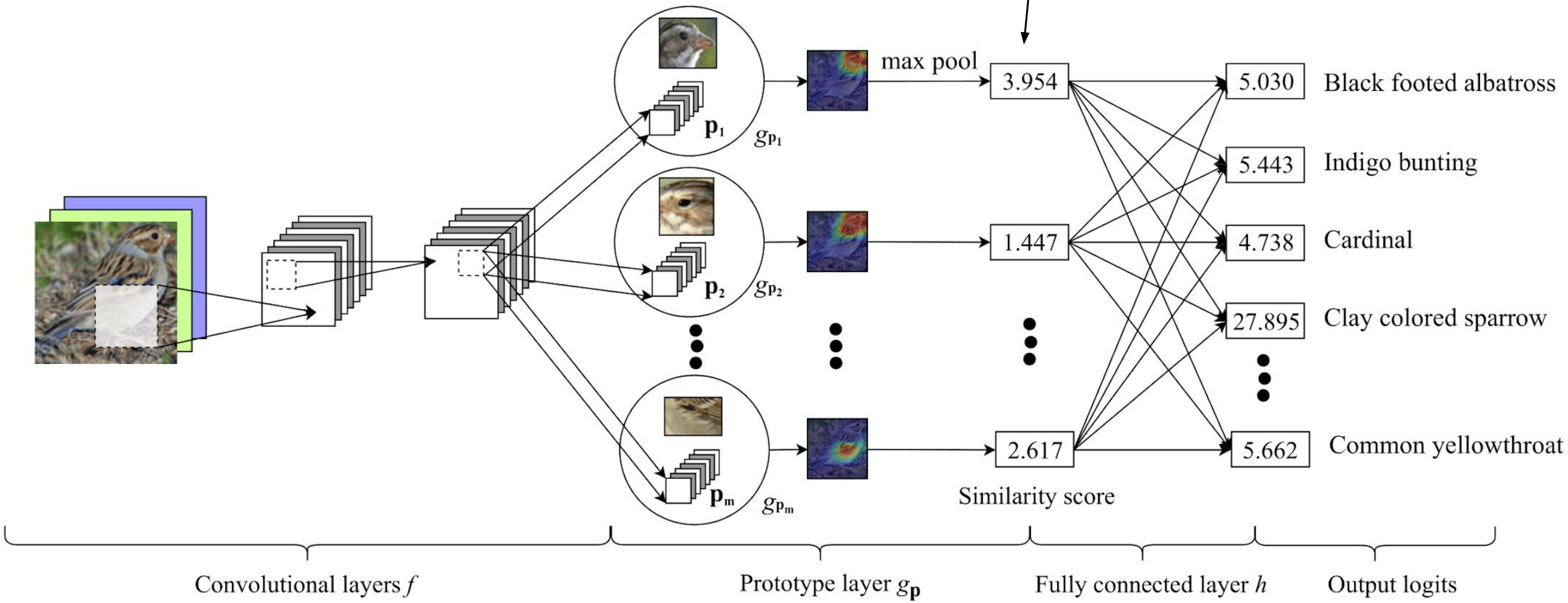
Leftmost: a test image of a clay-colored sparrow
Second column: same test image, each with a bounding box generated by our model -- the content within the bounding box is considered by our model to look similar to the prototypical part (same row, third column) learned by our algorithm
Third column: prototypical parts learned by our algorithm
Fourth column: source images of the prototypical parts in the third column
Rightmost column: activation maps indicating how similar each prototypical part resembles part of the test bird

Chen et al. This looks like that: deep learning for interpretable image recognition, NeurIPS 2019

Architecture

$$g(Z_x, p) = \max_{z \in Z_x} \log \left(\frac{\|z-p\|^2+1}{\|z-p\|^2+\varepsilon} \right) \quad \text{for} \quad \varepsilon > 0.$$

Monotonically decreasing with respect to $|z-p\|^2$



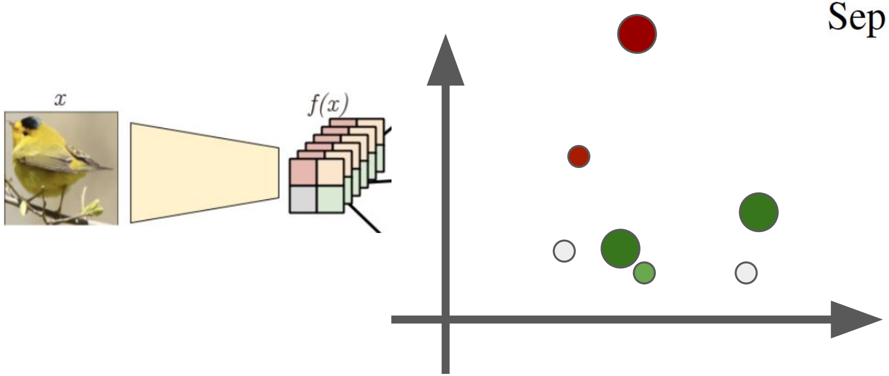
Training

- Training phases (warm-up, main, push, finetuning)
- Special loss function

$$\min_{\mathbf{P}, w_{\text{conv}}} \frac{1}{n} \sum_{i=1}^{n} \text{CrsEnt}(h \circ g_{\mathbf{p}} \circ f(\mathbf{x_i}), \mathbf{y_i}) + \lambda_1 \text{Clst} + \lambda_2 \text{Sep}$$

$$\text{Clst} = \frac{1}{n} \sum_{i=1}^{n} \min_{j: \mathbf{p}_j \in \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

$$\text{Sep} = -\frac{1}{n} \sum_{i=1}^{n} \min_{j: \mathbf{p}_j \notin \mathbf{P}_{y_i}} \min_{\mathbf{z} \in \text{patches}(f(\mathbf{x}_i))} \|\mathbf{z} - \mathbf{p}_j\|_2^2$$

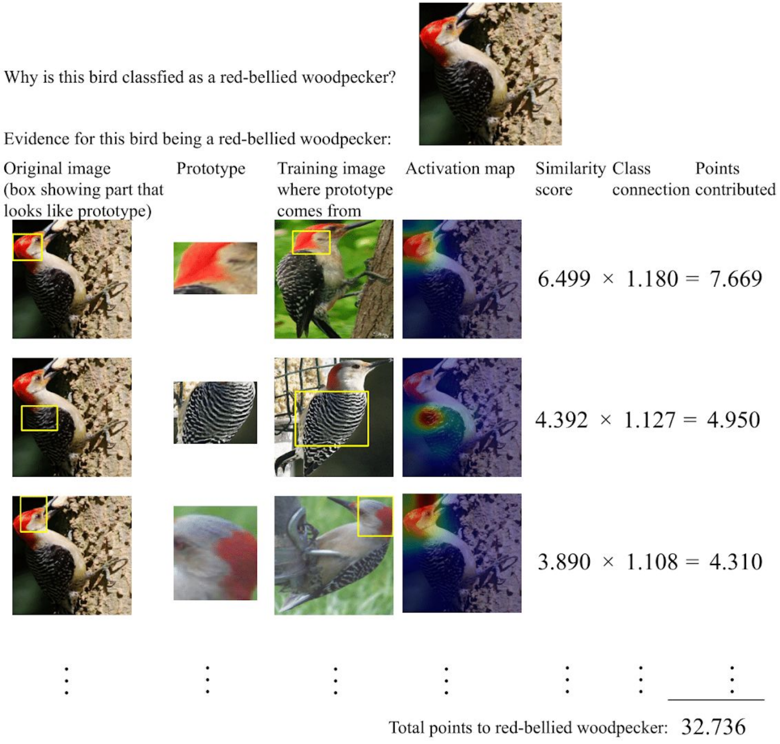


Experimental setup

- Tested on two datasets: CUB-200-2011 and Stanford Cars

Results

Interpretability	Model: accuracy
None	**B-CNN**[25]: 85.1 (bb), 84.1 (full)
Object-level attn.	**CAM**[56]: 70.5 (bb), 63.0 (full)
Part-level attention	**Part R-CNN**[53]: 76.4 (bb+anno.); **PS-CNN** [15]: 76.2 (bb+anno.); **PN-CNN** [3]: 85.4 (bb+anno.); **DeepLAC**[24]: 80.3 (anno.); **SPDA-CNN**[52]: 85.1 (bb+anno.); **PA-CNN**[19]: 82.8 (bb); **MG-CNN**[46]: 83.0 (bb), 81.7 (full); **ST-CNN**[16]: 84.1 (full); **2-level attn.**[49]: 77.9 (full); **FCAN**[26]: 82.0 (full); **Neural const.**[37]: 81.0 (full); **MA-CNN**[55]: 86.5 (full); **RA-CNN**[7]: 85.3 (full)
Part-level attn. + prototypical cases	**ProtoPNet** (ours): 80.8 (full, VGG19+Dense121+Dense161-based) 84.8 (bb, VGG19+ResNet34+DenseNet121-based)

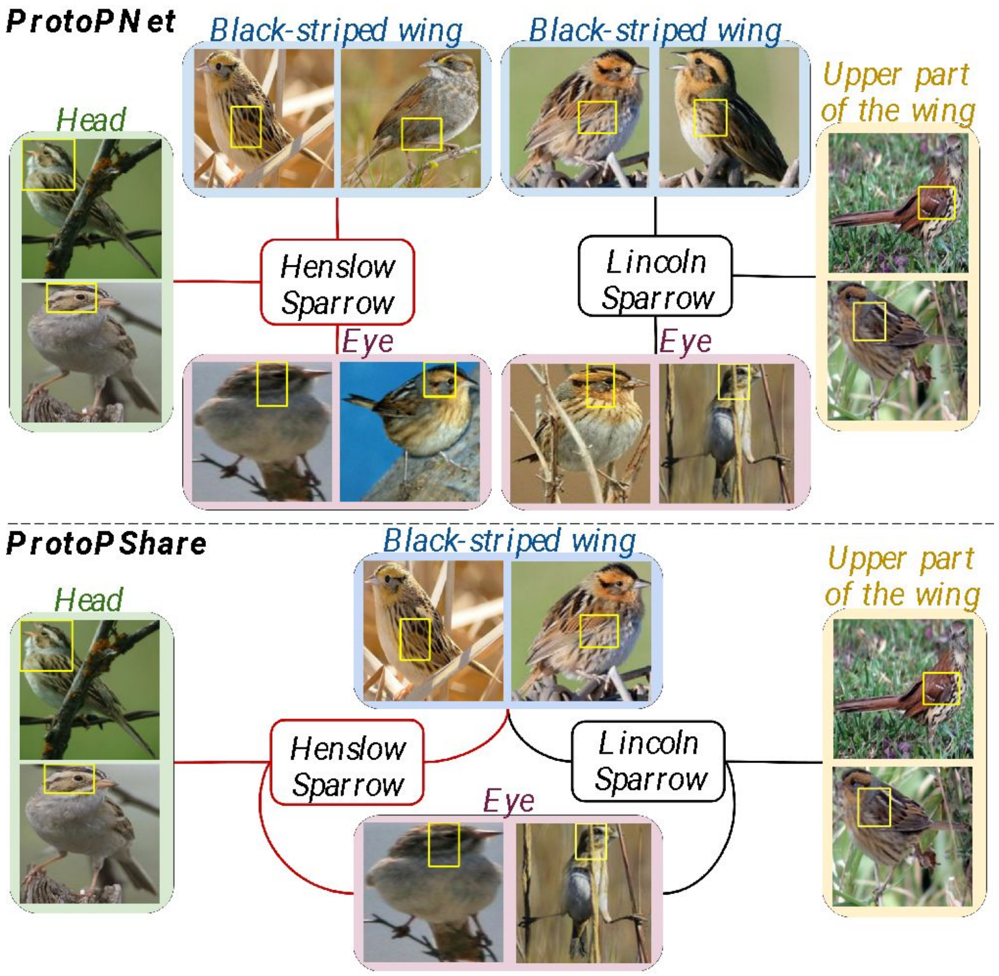


ProtoPNet limitations

- Large number of prototypes (each of them is assigned to only one class)
- Similar prototypes of two different classes can be distant in representation space (here, fender)

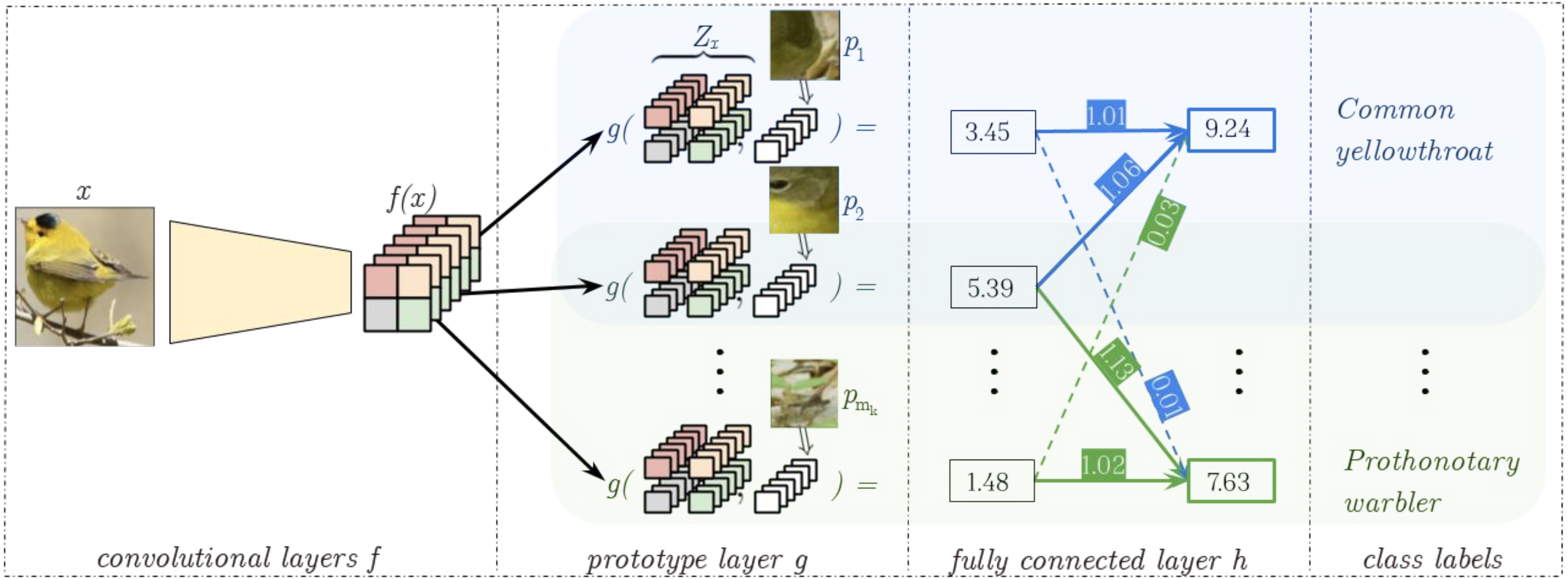
ProtoPShare

Idea



Rymarczyk et al. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. KDD 2021

Architecture

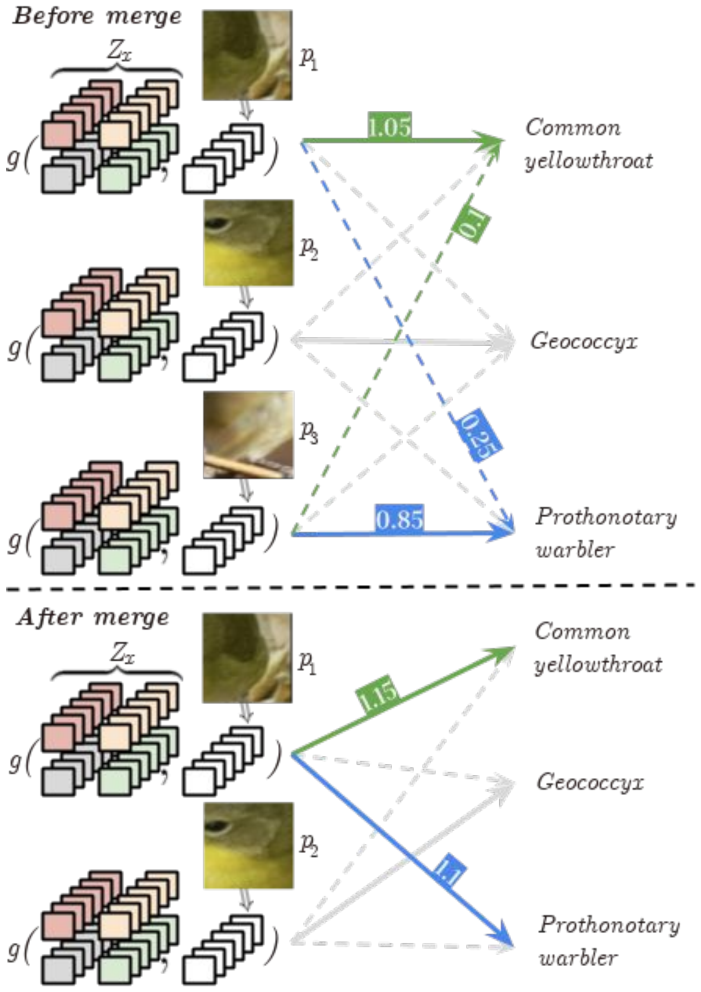


Algorithm

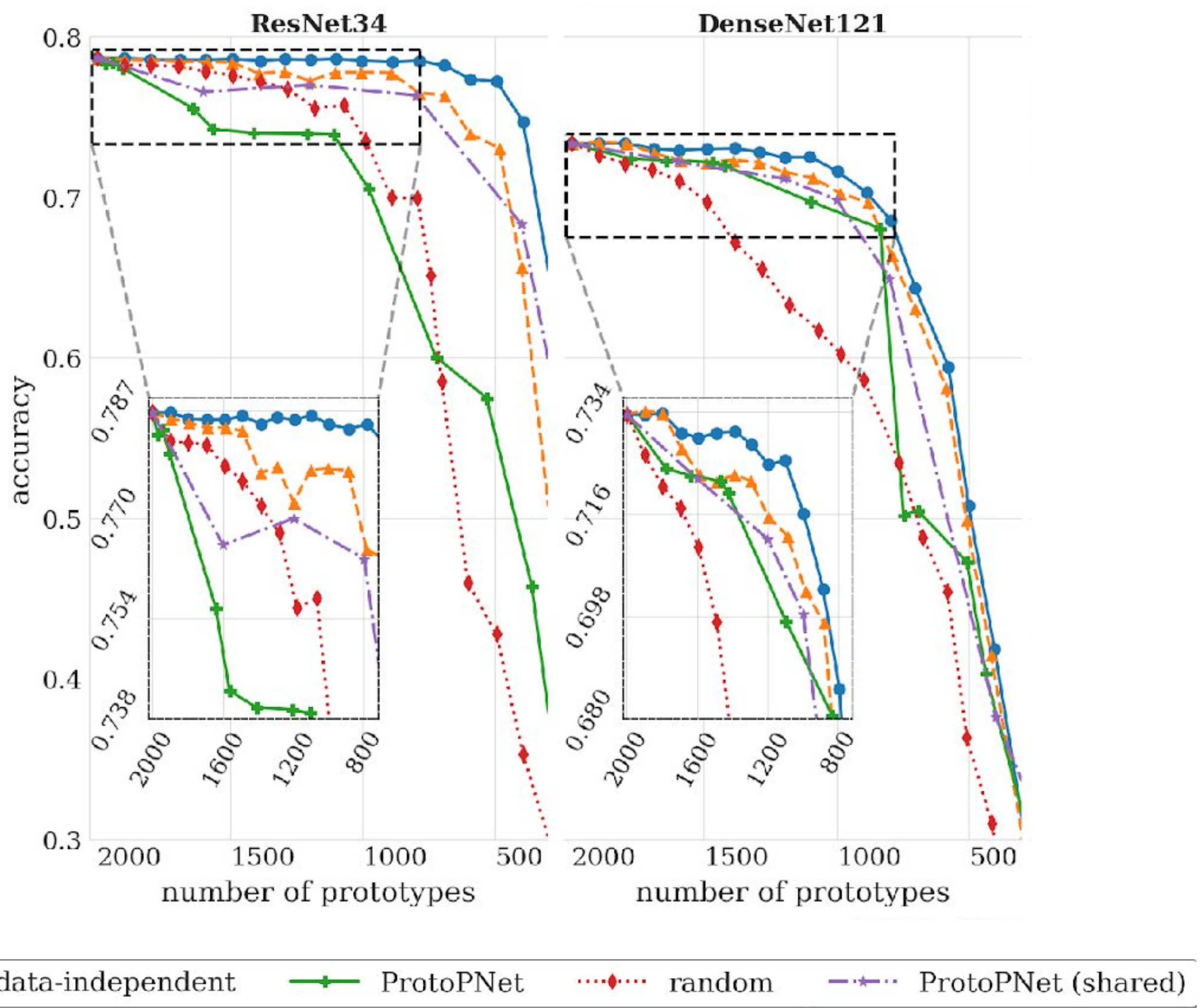
- Run ProtoPNet with standard settings
- Repetively find the two most similar prototypes and merge them into one
- Use data-dependent similarity, where prototypes are considered similar if they activate alike on the training images:

$$d_{DD}(p, \tilde{p}) = \frac{1}{\sum_{x \in X} \left(g(Z_x, p) - g(Z_x, \tilde{p})\right)^2}$$

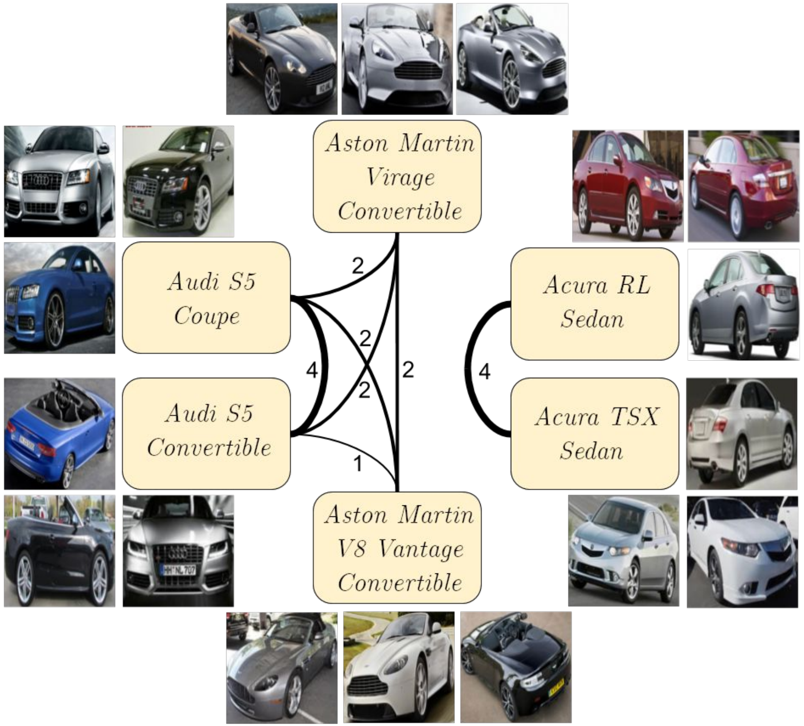
$$g(Z_x, p) = \max_{z \in Z_x} \log \left(\frac{\|z-p\|^2+1}{\|z-p\|^2+\varepsilon} \right) \quad \text{for} \quad \varepsilon > 0.$$

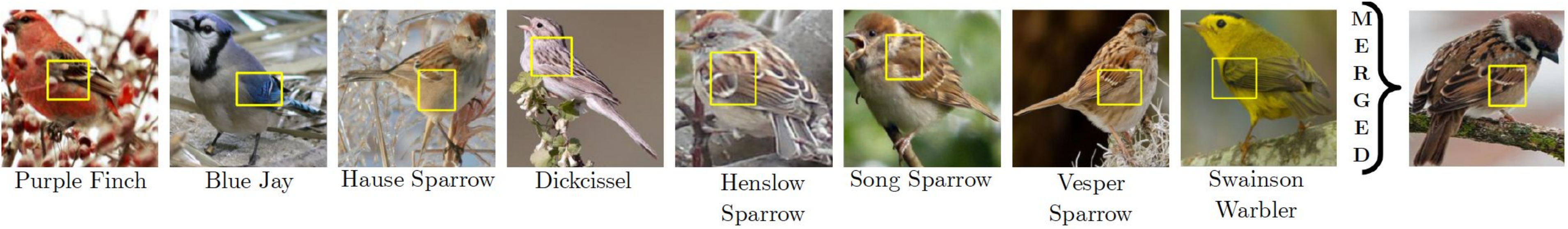


ProtoPShare



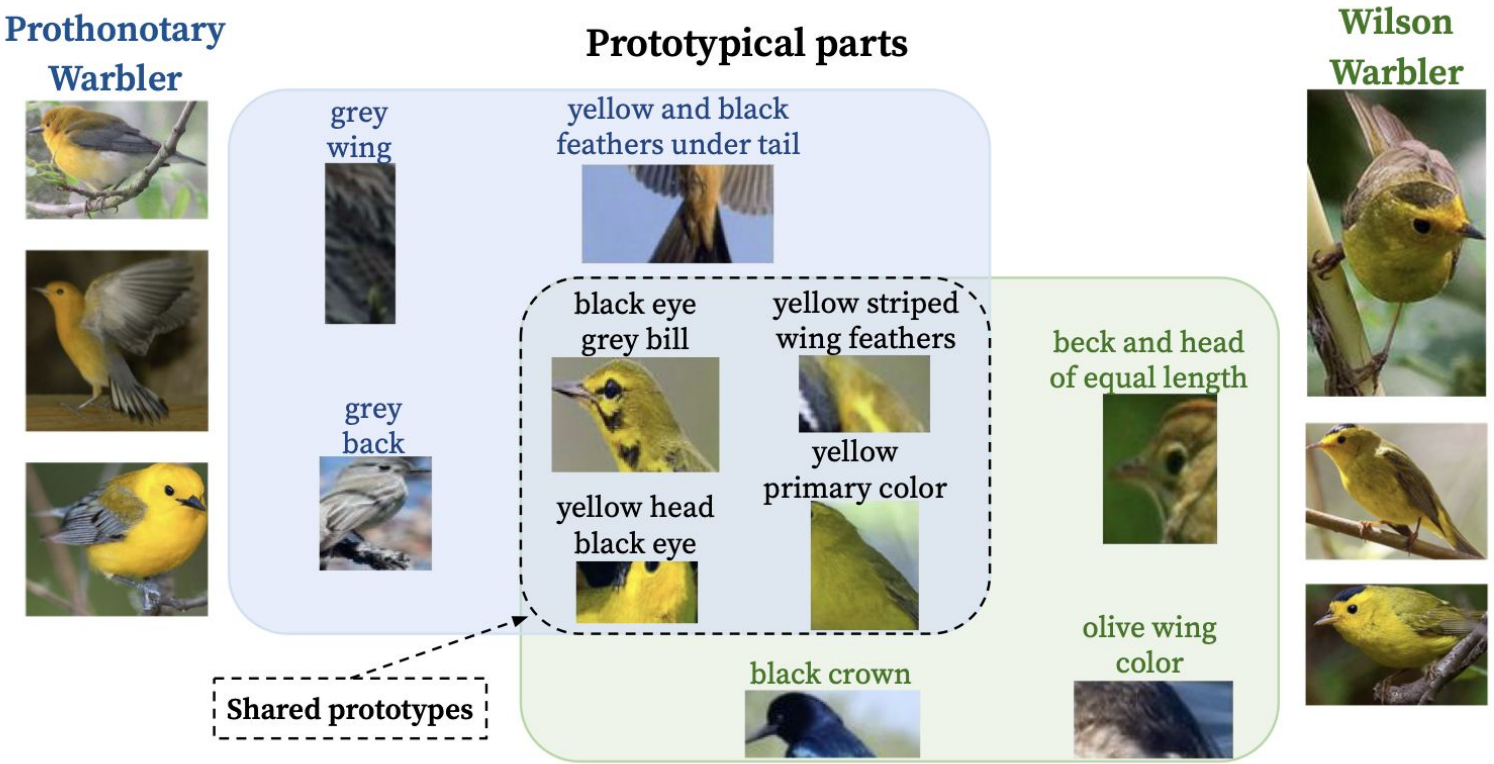
ProtoPShare advantages





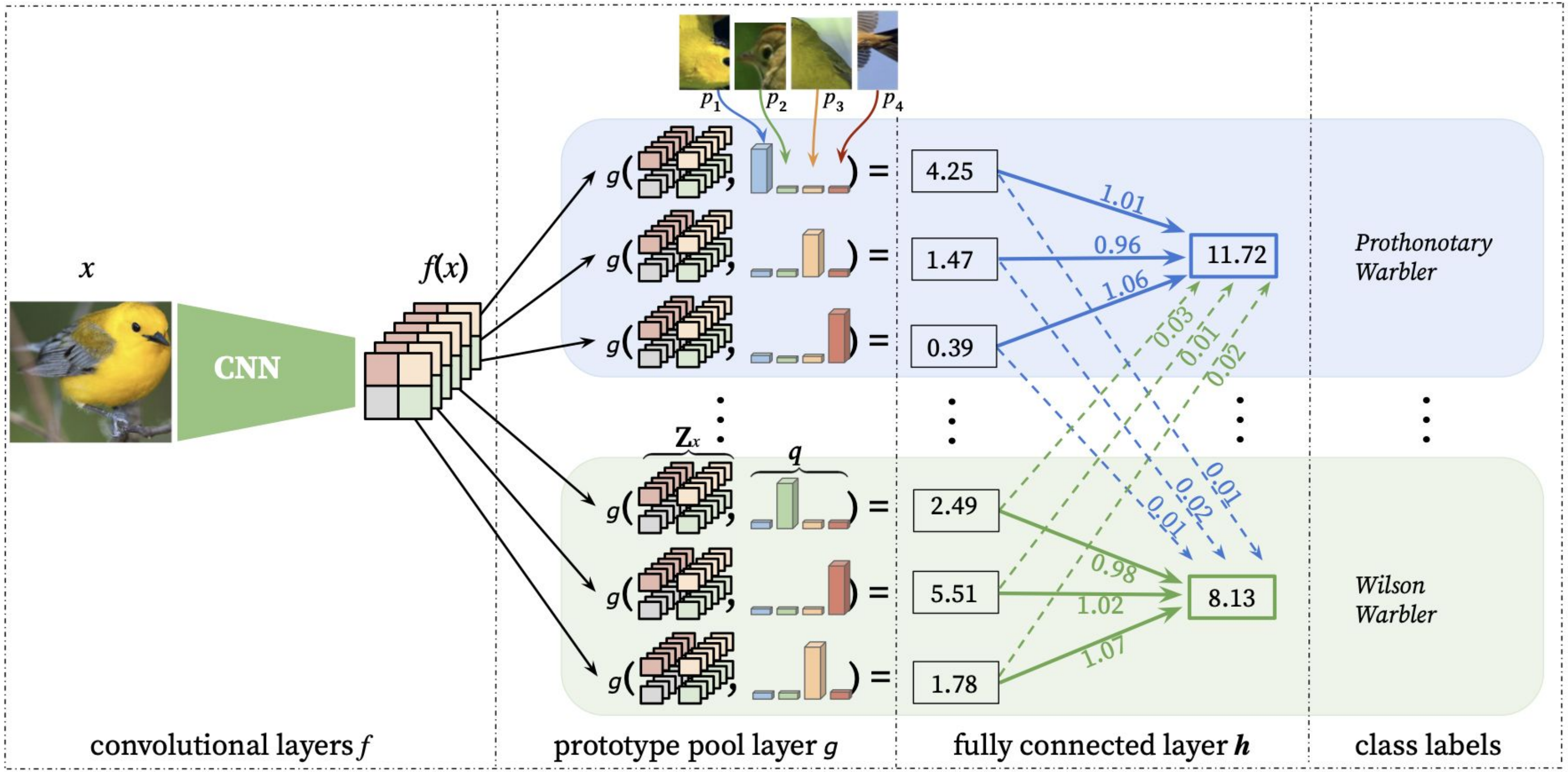
ProtoPool

Idea



Rymarczyk et al. Interpretable Image Classification with Differentiable Prototypes Assignment. ECCV 2022

Architecture



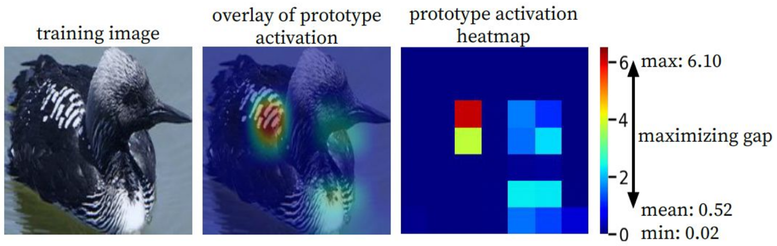
Training

- We use the Gumbel-Softmax trick to learn the assignments of prototypes to data classes: $\text{Gumbel-softmax}(q, \tau) = (y^1, \dots, y^M) \in \mathbb{R}^M$

$$y^i = \frac{\exp\big((q^i + \eta_i)/\tau\big)}{\sum_{m=1}^{M} \exp((q^m + \eta_m)/\tau)}$$

- We introduce a focal similarity function that widens the gap between maximal and average activation:

$$g_p = \max_{z \in Z_x} g_p(z) - \operatorname*{mean}_{z \in Z_x} g_p(z)$$



- Force different prototypes in different slots of the same class:

$$\mathcal{L}_{orth} = \sum_{i<j}^{K} \frac{\langle q_i, q_j \rangle}{\|q_i\|_2 \cdot \|q_j\|_2}$$

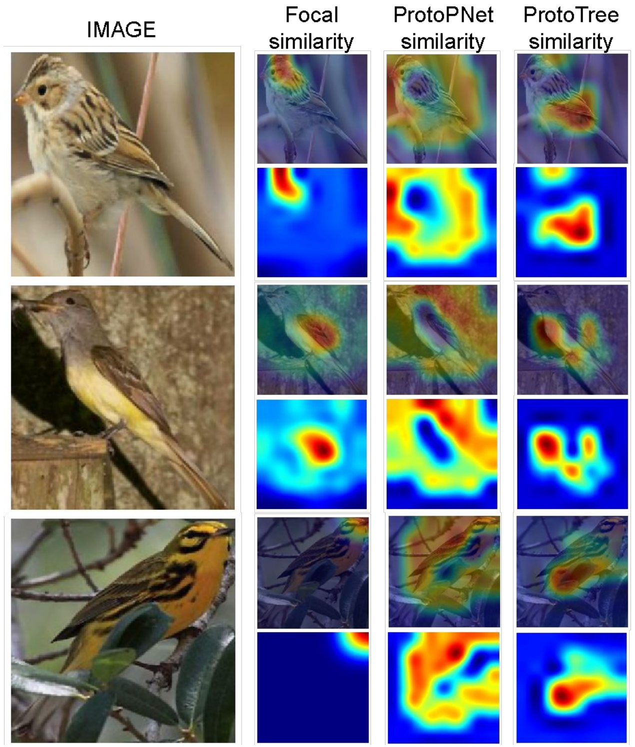
Results

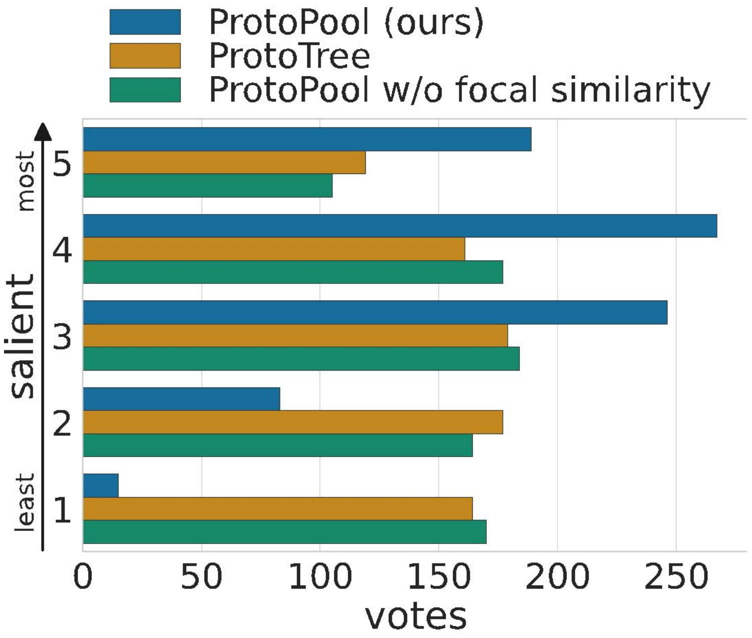
CUB-200-2011			
Model	Arch.	Proto. #	Acc [%]
ProtoPool (ours)	R34	202	80.3±0.2
ProtoPShare [47]		400	74.7
ProtoPNet [8]		1655	79.5
TesNet [56]		2000	82.7±0.2
ProtoPool (ours)	R152	202	81.5±0.1
ProtoPShare [47]		1000	73.6
ProtoPNet [8]		1734	78.6
TesNet [56]		2000	82.8±0.2
ProtoPool (ours)	iNR50	202	85.5±0.1
ProtoTree [38]		202	82.2±0.7
ProtoPool (ours)	Ex3	202×3	87.5
ProtoTree [38]		202×3	86.6
ProtoPool (ours)	Ex5	202×5	**87.6**
ProtoTree [38]		202×5	87.2
ProtoPNet [8]		2000×5	84.8
TesNet [56]		2000×5	86.2

Stanford Cars			
Model	Arch.	Proto. #	Acc [%]
ProtoPool (ours)	R34	195	89.3±0.1
ProtoPShare [47]		480	86.4
ProtoPNet [8]		1960	86.1±0.2
TesNet [56]		1960	92.6±0.3
ProtoPool (ours)	R50	195	88.9±0.1
ProtoTree [38]		195	86.6±0.2
ProtoPool (ours)	Ex3	195×3	91.1
ProtoTree [38]		195×3	90.5
ProtoPool (ours)	Ex5	195×5	91.6
ProtoTree [38]		195×5	91.5
ProtoPNet [8]		1960×5	91.4
TesNet [56]		1960×5	**93.1**

Model	ProtoPool	ProtoTree	ProtoPShare	ProtoPNet	TesNet
Portion of prototypes	∼10%	∼10%	[20%;50%]	100%	100%
Reasoning type	+	+/−	+	+	+
Prototype sharing	direct	indirect	direct	none	none

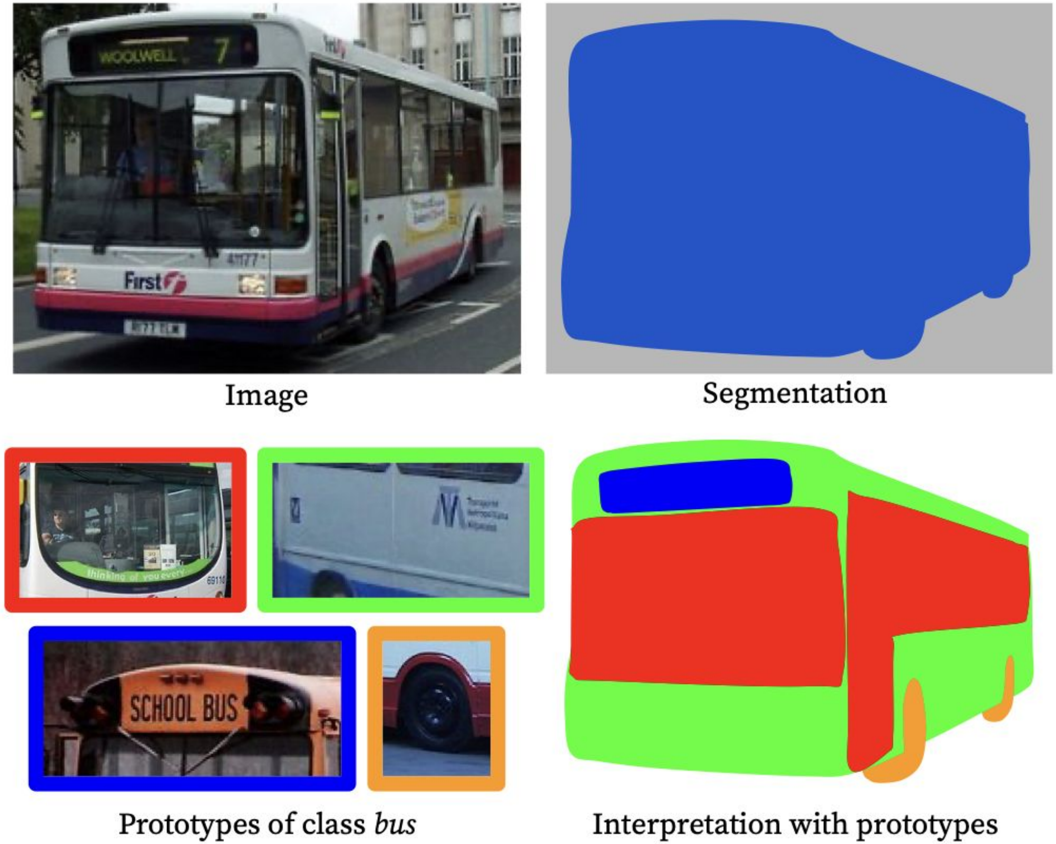
User studies





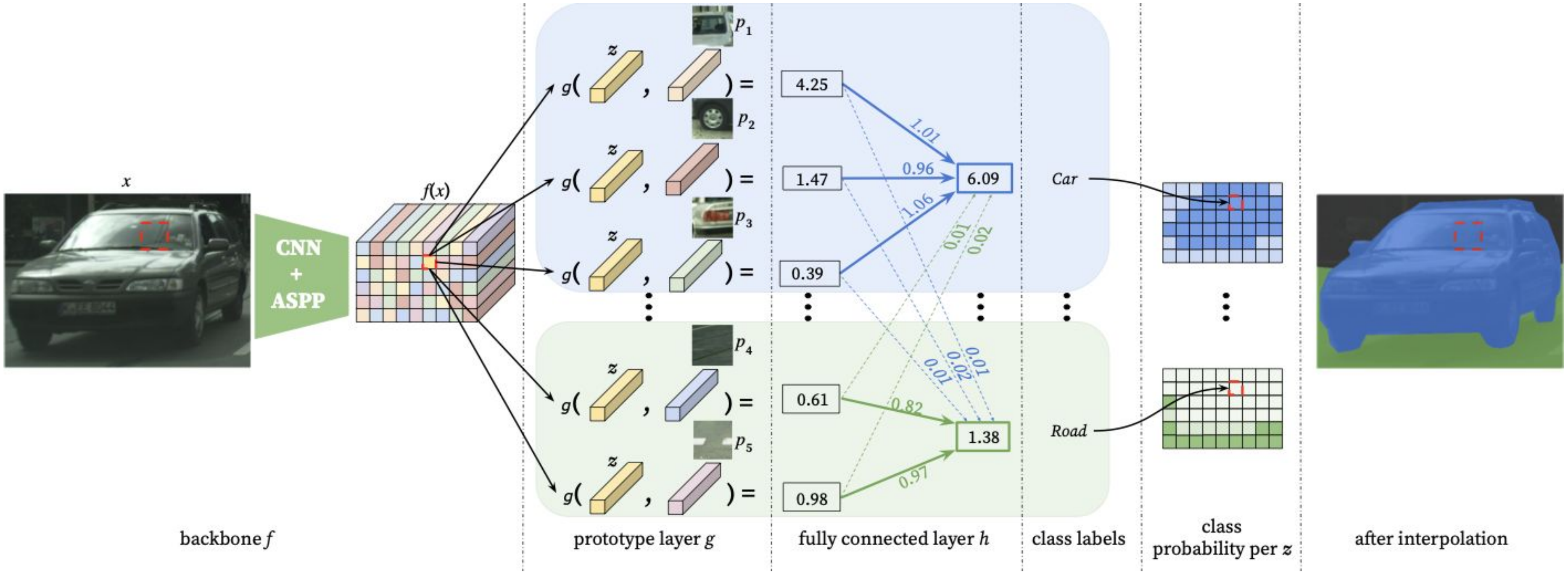
ProtoSeg

Idea



Sacha et al. ProtoSeg: Interpretable Semantic Segmentation with Prototypical Parts. WACV 2023

Architecture



Training

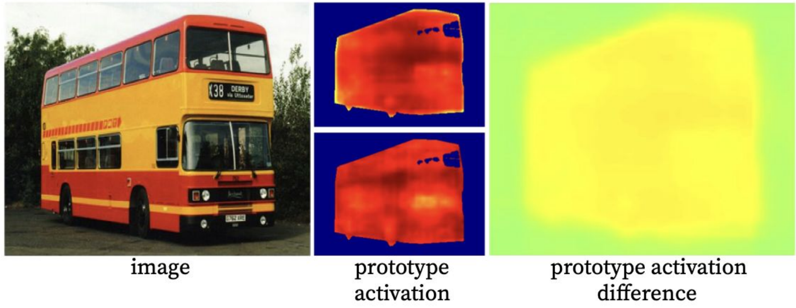
- Differentiate prototypes of same class using Jeffrey's similarity:

$$\mathcal{L}_{\text{J}}(Z, \mathbf{P}_c) = \mathcal{S}_J(v(Z, p_1), \ldots, v(Z, p_k))$$

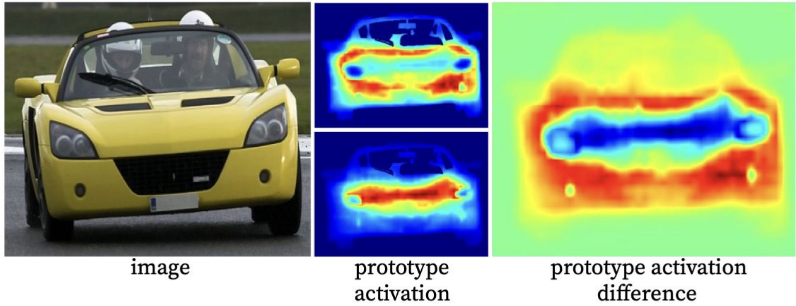
$$\mathcal{S}_J(U_1, \ldots, U_l) = \frac{1}{\binom{l}{2}} \sum_{i<j} \exp(-\mathcal{D}_J(U_i, U_j))$$

$$\mathcal{D}_J(U, V) = \tfrac{1}{2}\mathcal{D}_{KL}(U \| V) + \tfrac{1}{2}\mathcal{D}_{KL}(V \| U)$$

$$v(Z, p) = \operatorname{softmax}(\|z_{ij} - p\|^2 \mid z_{ij} \in Z : Y_{ij} = c)$$



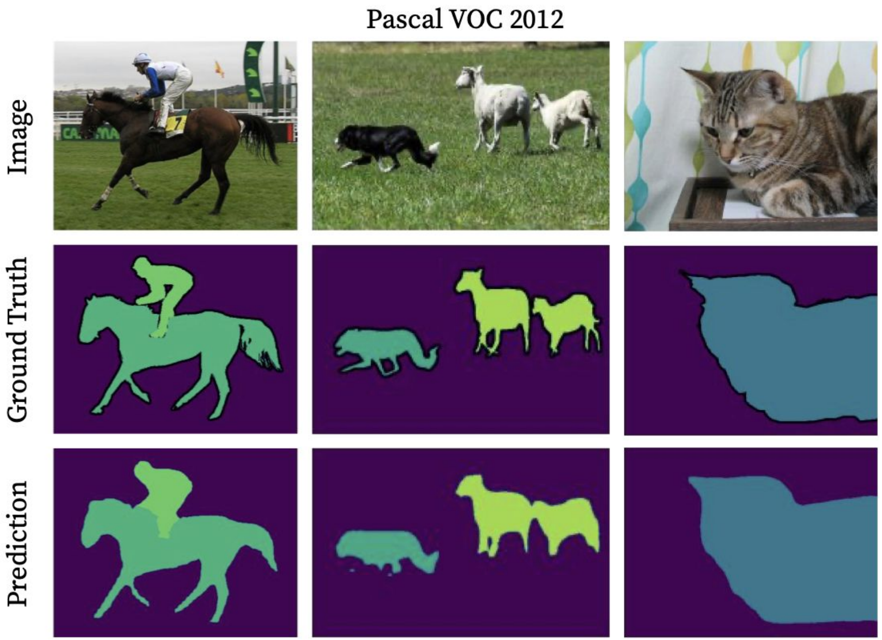
(a) High value of $\mathcal{L}_{\text{J}}$.



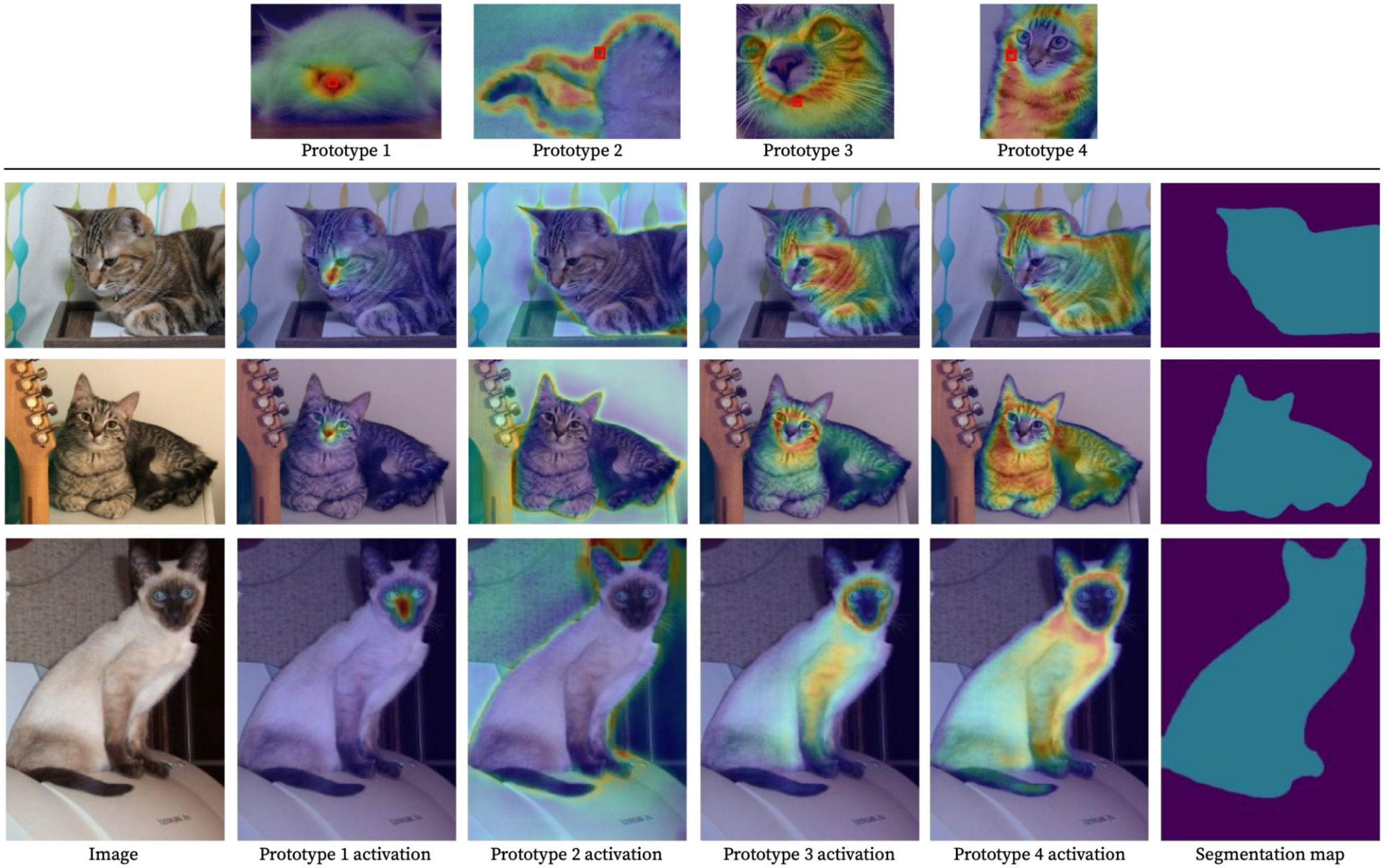
(b) Low value of $\mathcal{L}_{\text{J}}$.

Results

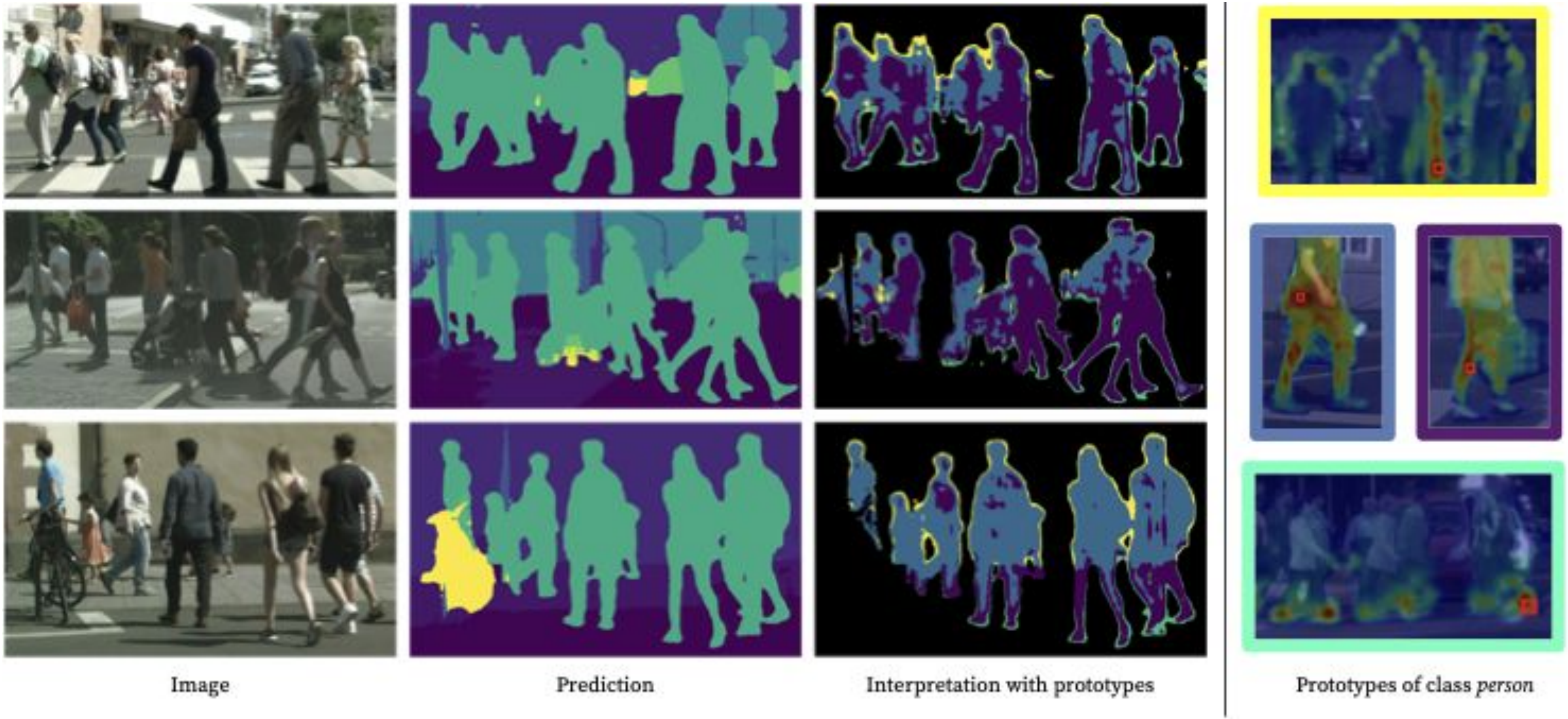
Dataset	Method	Pretraining	mIOU val	mIOU test
Pascal	DeepLabv2	COCO	77.69	79.70
	ProtoSeg	COCO	67.98	68.71
	ProtoSeg	ImageNet	72.05	72.92
Cityscapes	DeepLabv2	COCO	71.40	70.40
	ProtoSeg	COCO	55.35	56.77
	ProtoSeg	ImageNet	67.23	67.04



Example (class cat)



Example (class person)



Conclusions & future works

Conclusions

- We provide self-explainable methods based on prototypes
- In contrast to existing methods, they:
  - share prototypes between classes
  - increase model interpretability
  - can be used to find similarities between classes
  - focus the model on salient features

Future works

- Sustainable and interpretable deep learning
- Interpretable counterfactual examples
- Prototypes (personalized) visualization
- Interactive interpretable learning

Thank you for your attention!

MLSS^S 2023

MACHINE LEARNING SUMMER SCHOOL
ON APPLICATIONS IN SCIENCE

26 JUNE - 2 JULY / KRAKÓW, POLAND