JAGIELLONIAN UNIVERSITY IN KRAKÓW

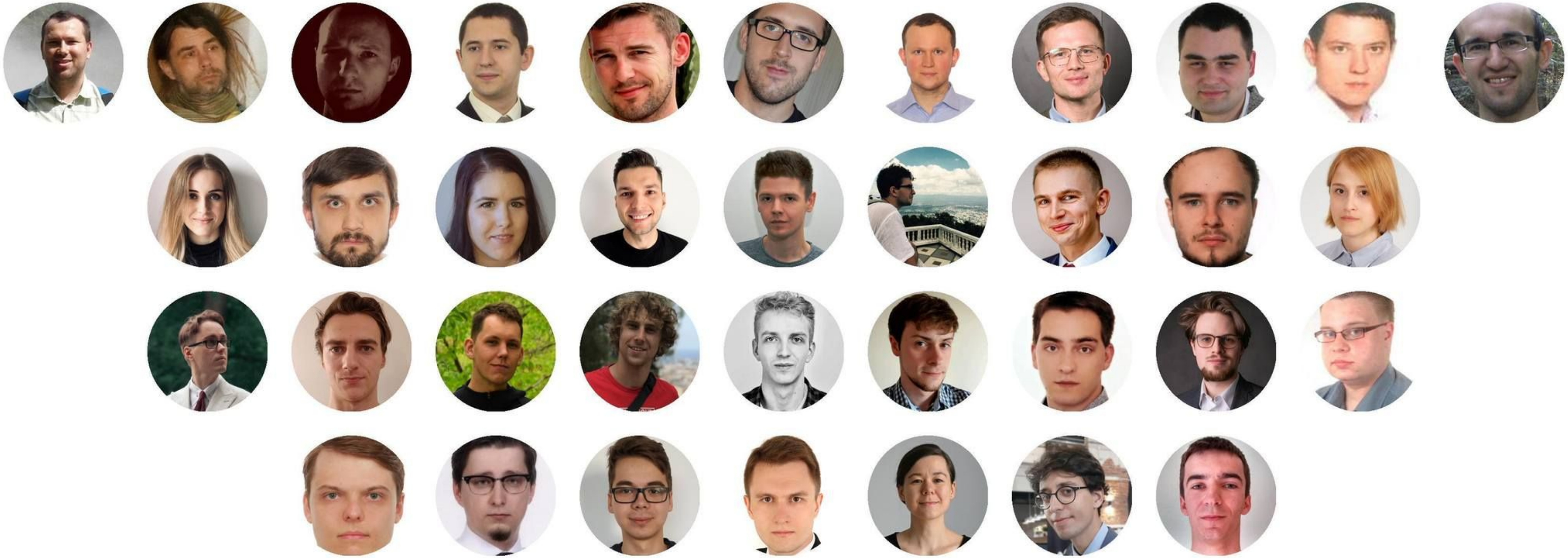
group of machine learning research gmum

Interpretability and explainability of deep models

Łukasz Struski

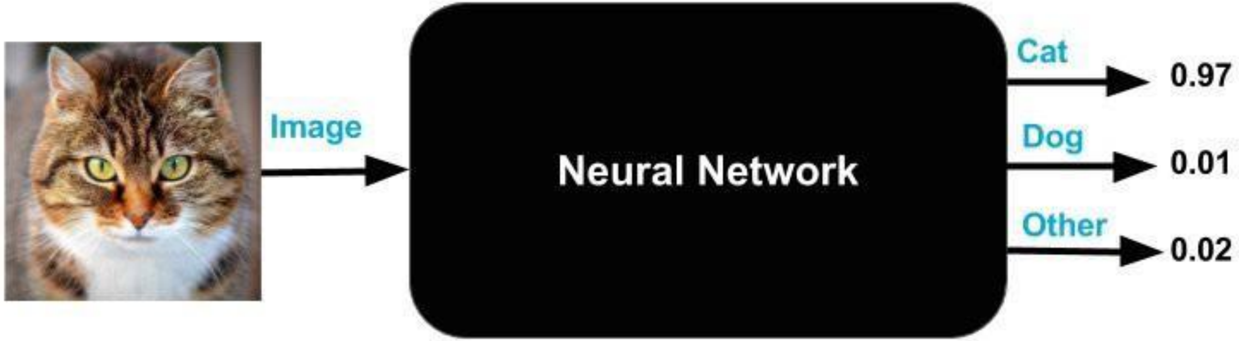
Team – https://gmum.net

group of machine
gmum
learning research



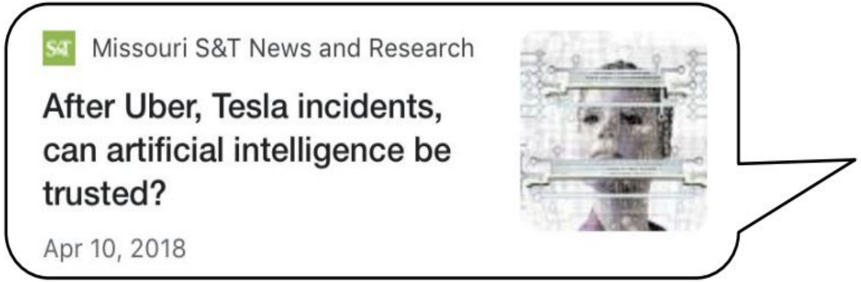
Motivation

- 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)



Motivation

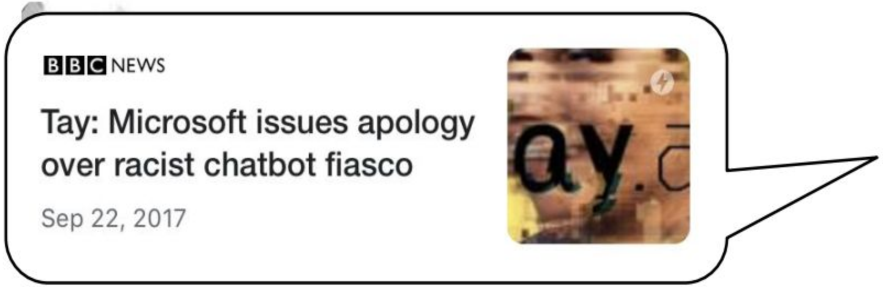
Wrong decisions can be costly and dangerous



Missouri S&T News and Research

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

Apr 10, 2018



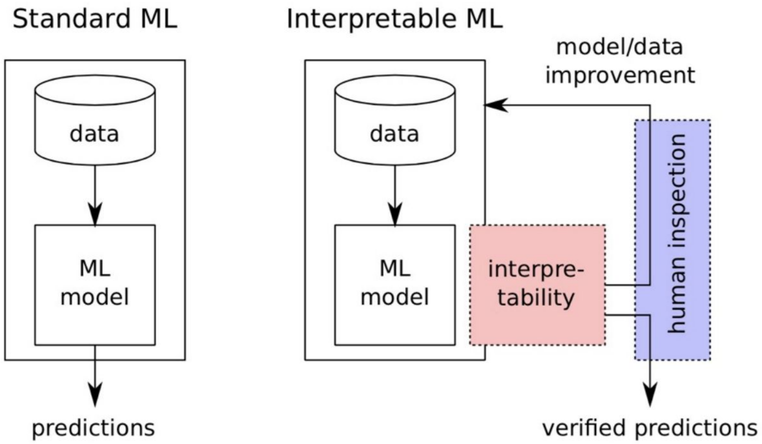
BBC NEWS

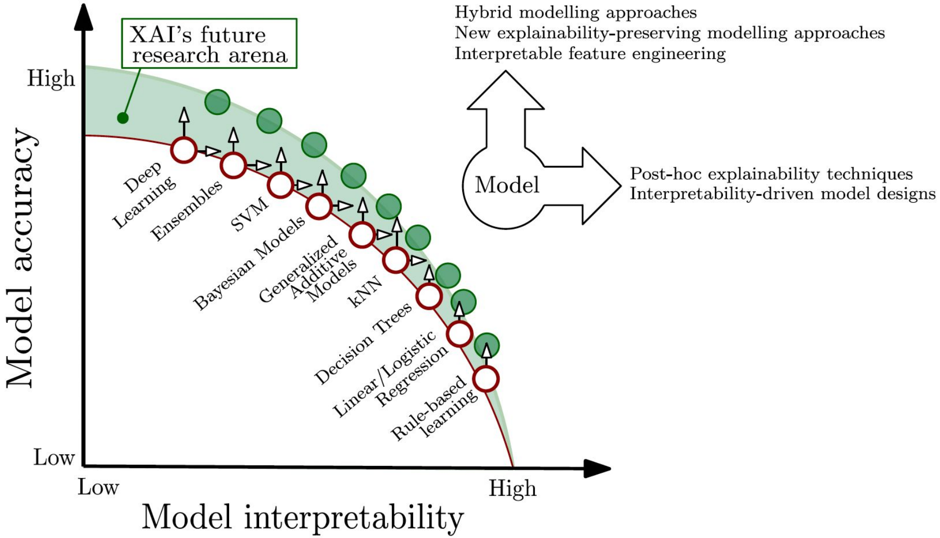
Tay: Microsoft issues apology over racist chatbot fiasco

Sep 22, 2017

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, 202

Division of the XAI world – Post-hoc methods

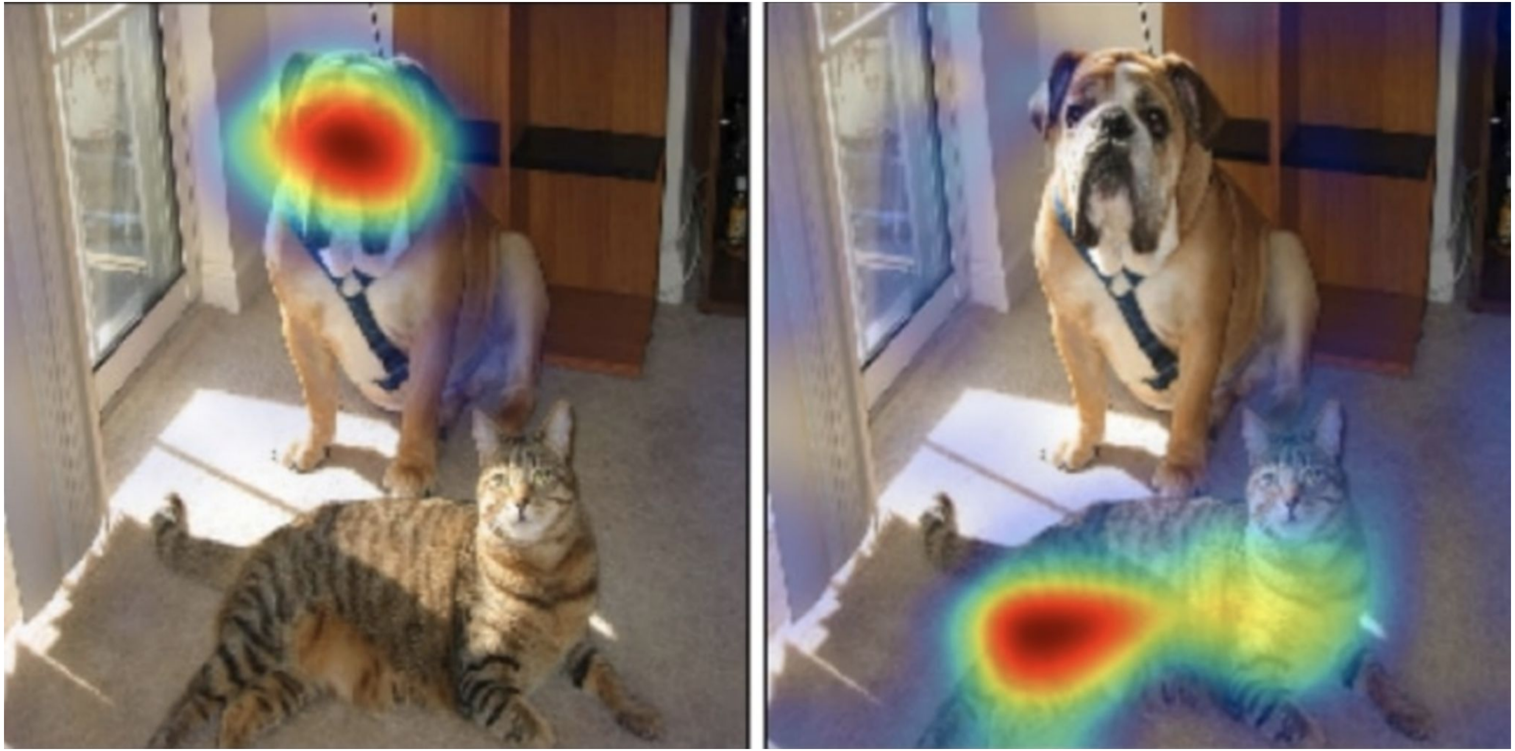
The aim is to explain the decision of a pre-trained network (could be black-box).

- Grad-Cam(s)
- Lime
- Shap Values
- etc

Typically what we can obtain is heat-map of the important features.

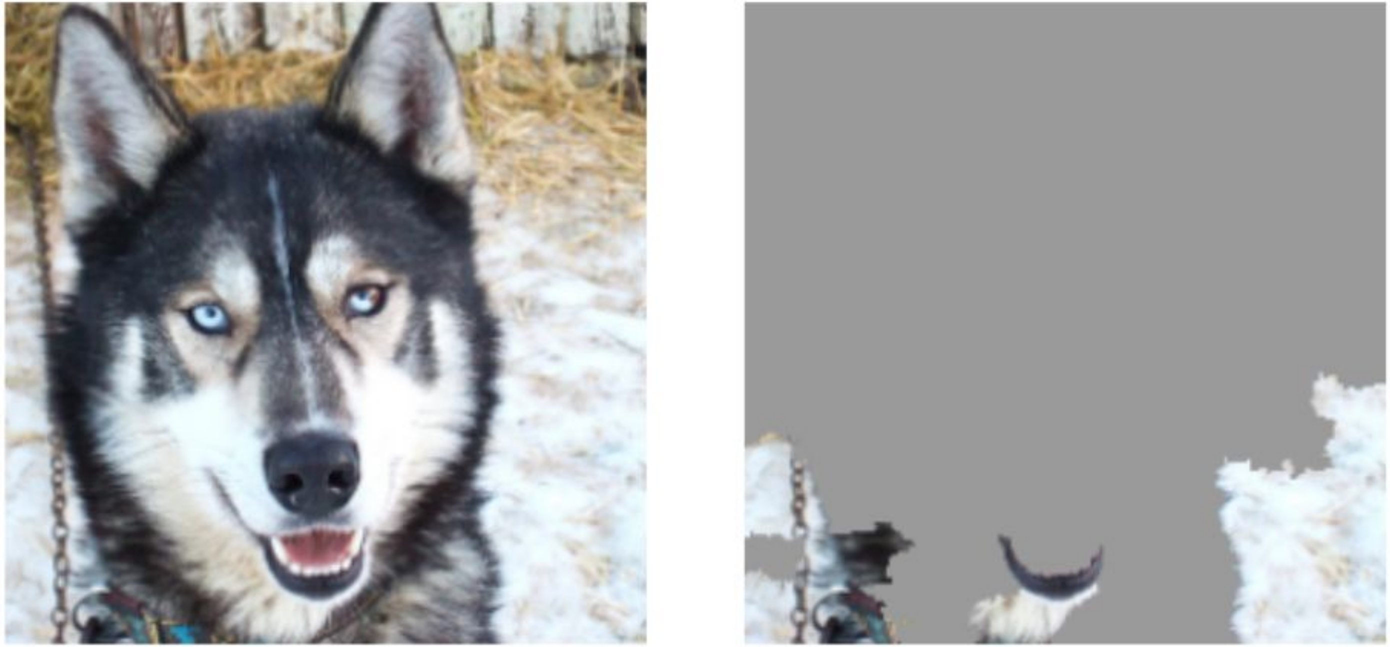
Consequently, we only know on which features the model focuses its attention.

Post-hoc methods: Grad-CAM



https://arxiv.org/pdf/1610.02391

Post-hoc methods: Why should I trust you?



(a) Husky classified as wolf

(b) Explanation

https://arxiv.org/abs/1602.04938

Division of the XAI world – Intrinsic models

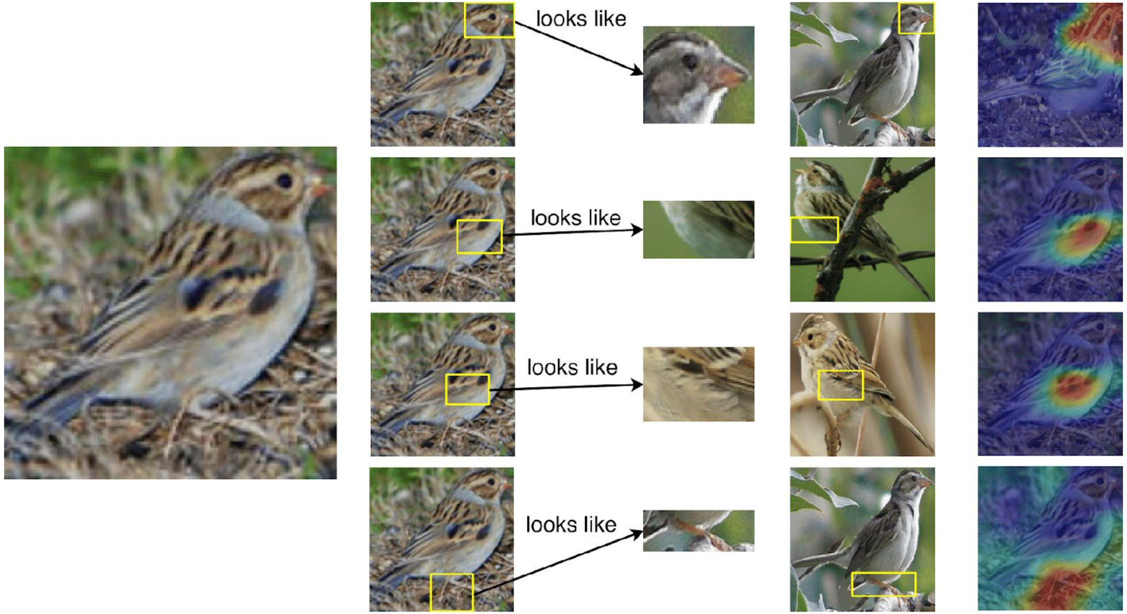
The aim is to construct models which decisions are possible to explain/understand.

- Decision trees
- Prototypical parts models
- b-cos networks

Typical disadvantages: harder to train, there often appears some cost of accuracy, often restricted to some datasets (prototypes), needs new architecture and therefore not always possible to fine-tune from existing models (b-cos networks).

Prototypical parts models
ProtoPNet: This looks like that

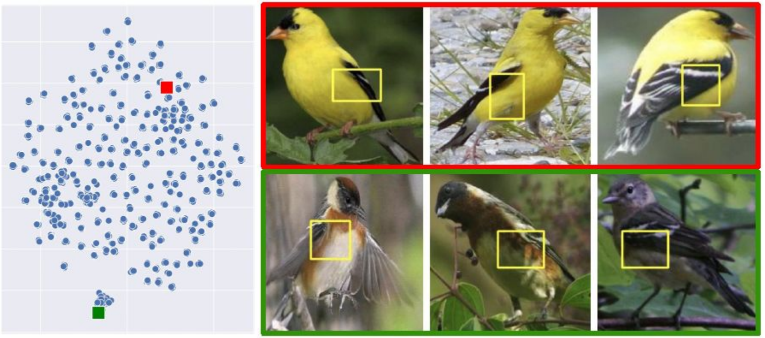
Aim of prototypical parts models is to create models that: **compare the input to reference patterns, represented by training data patches**.

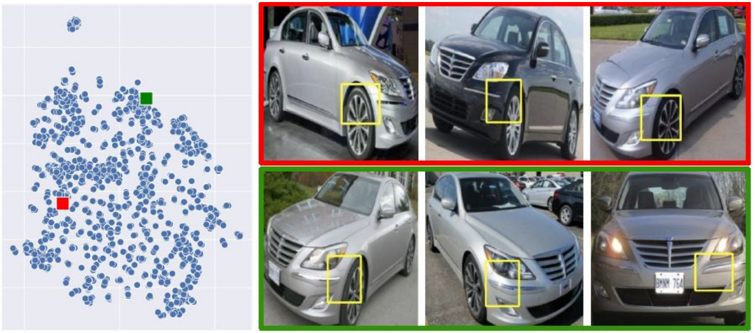


Leftmost: a test image of a clay-colored sparrow
Second column: same test image, each with a bounding box generated by our model -- the content within the bounding box is considered by our model to look similar to the prototypical part (same row, third column) learned by our algorithm
Third column: prototypical 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

Prototypical parts models – 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, bright belly with grayish wings and fender)





Prototypical parts models – our contribution

- **Rymarczyk, Struski, Tabor and Zieliński.** ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In ACM International Conference on Knowledge Discovery and Data Mining (KDD), 2021 – arxiv.org/pdf/2011.14340
- **Rymarczyk, Struski, Górszczak, Lewandowska, Tabor and Zieliński.** Interpretable Image Classification with Differentiable Prototypes Assignment. In European Conference on Computer Vision (ECCV), 2022 – arxiv.org/pdf/2112.02902
- **Sacha, Jura, Rymarczyk, Struski, Tabor, Zieliński.** Interpretability Benchmark for Evaluating Spatial Misalignment of Prototypical Parts Explanations. National Conference of the American Association for Artificial Intelligence (AAAI), 2024 – arxiv.org/pdf/2308.08162

Prototypical parts models – our contribution

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
- interpretability benchmarks

Prototypical parts models

Pros

- we can really (hope to) understand the decision of prototypical networks (contrary to Post-hoc methods where we have only attention of the network!)
- we have disentangled the final decision into simpler atomic components, where each can be easier to understand.

Cons

- we have to construct new architecture and loss functions
- the training can be nontrivial
- since prototypes look at the local differences, it works well for homogeneous classes (birds/dogs/cars) but does not work well on ImageNet since the classes or not mutations of some one main general class

XAI world is broken!

Post-hoc methods

- We cannot explain the reasons behind the decisions of convolutional or transformer networks. We can only see where the network focuses its attention.

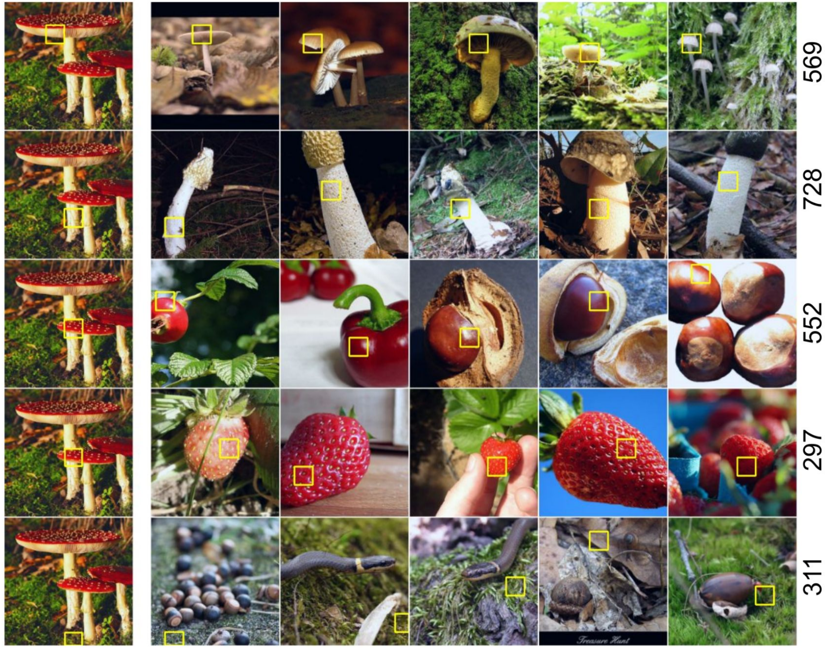
Inherently explained models

- To understand the decisions we need to construct special networks/architectures and loss functions.

InfoDisent: hybrid model

- We want to understand decisions (representation space) of pretrained convolutional or transformer networks.
- Motivated by prototypical parts networks, we aim to disentangle final decisions into understandable atomic components.
- Each atomic component will be represented by (prototypical) channel in the representation layer.
- Positive reasoning.

Input (Agaric)



Prototypes

https://arxiv.org/pdf/2409.10329

Research hypothesis: channels are not informative

The NN does not have any incentive to disentangle the information between channels (disentanglement means that the channels give information which is independent).

Let I denote the input image pushed through NN to the representation layer (last layer before the head), with k channels. Standard classification head is given by

$$class(I)=softmax\ A(avg\_pool(I)),$$

where *avg_pool* is taken over channels.

Research hypothesis: channels are not informative

Observation: operation *avg_pool* (applied channelwise) is commutative operation with matrix operations applied pixelwise:

$$avg\_pool(U_{pixelwise} I) = U\ avg\_pool(I).$$

Consequently: For any invertible matrix *U* we have the equality

$$softmax\ A\ avg\_pool(I) = softmax(AU^{-1})\ avg\_pool(U_{pixelwise} I).$$

This means that the one can mix the channels arbitrarily with invertible matrix, and unmix in the last linear layer, and obtain exactly the same result.

Two components of InfoDisent

Since we agree, that the channels contain entangled/mixed information, the appears to questions:

- which class of invertible matrices use for unmixing?
- how to devise an unmixing/disentangling mechanics? In other words how to motivate the network so that it would make the channels independent?

Unmixing by orthogonal matrices

As we have shown, we can theoretically unmix by any invertible matrix. We have decided to restrict to orthogonal matrices (isometries), as they do not change the innerlying scalar product. Recall that a square matrix U is orthogonal if $U^TU=UU^T=I$.

We even restrict it further to those which do not change orientation ($det=1$).

Parametrization – an arbitrary orthogonal matrix with $det=1$ is given by matrix exponential of skew symmetric matrix. Thus

$$U=exp(A-A^T),$$

where A is an arbitrary square matrix.

Information bottleneck

In the standard classification head we have the *avg_pool* operations, which aggregates/uses information from all pixels in the given channel.

Sparse Pooling Layer: We restrict this and construct a new pooling mechanism called *mx_pool* which in the pooling will have access to only two pixels, the largest positive and smallest negative.

$$mx\_pool(K)=max(ReLu(K)) - max(ReLu(-K)).$$

Final model

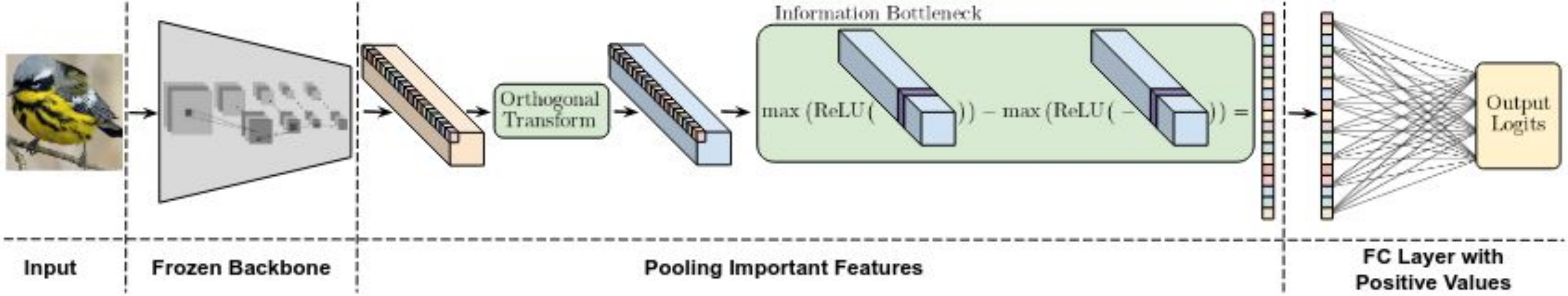
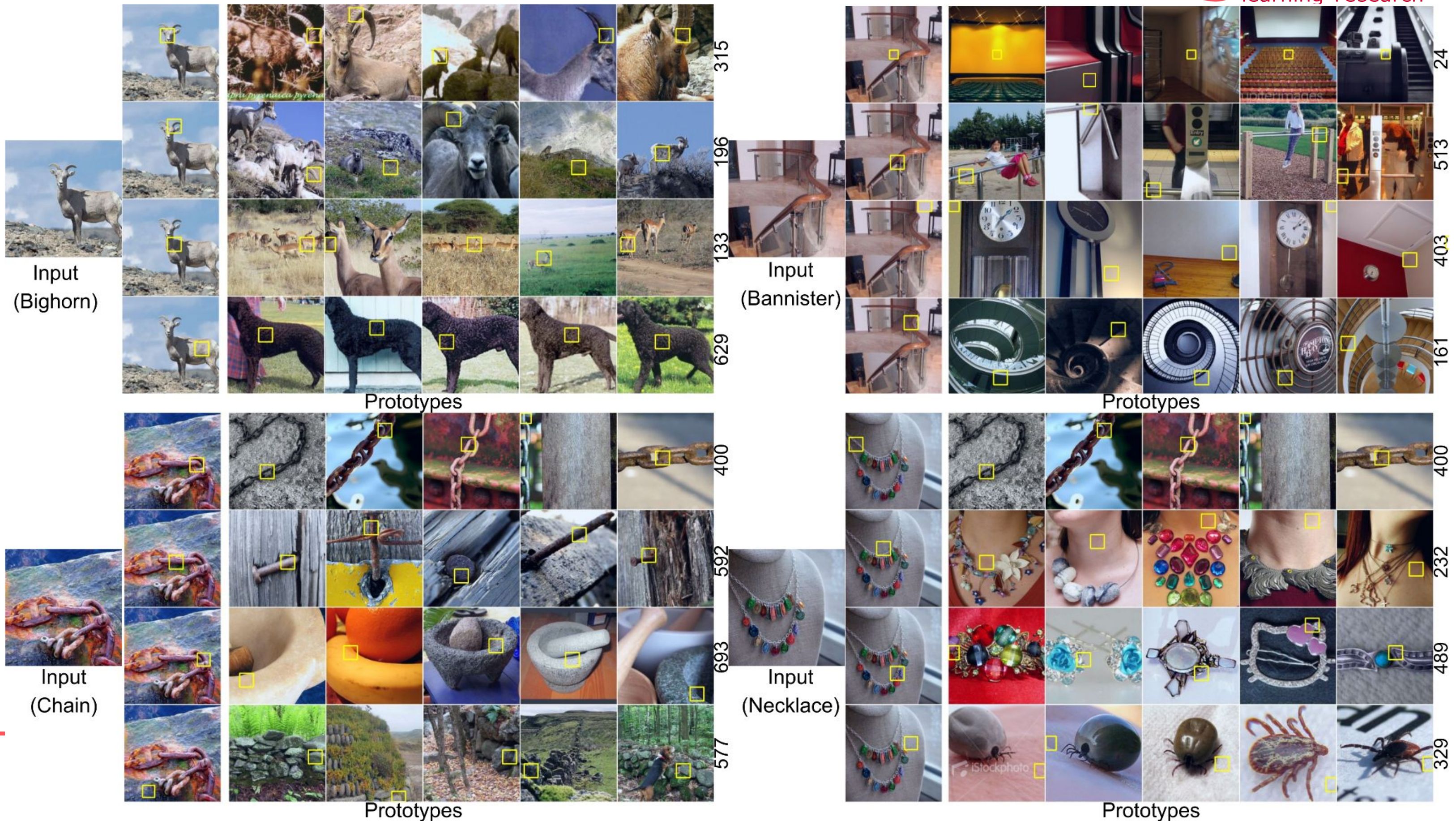
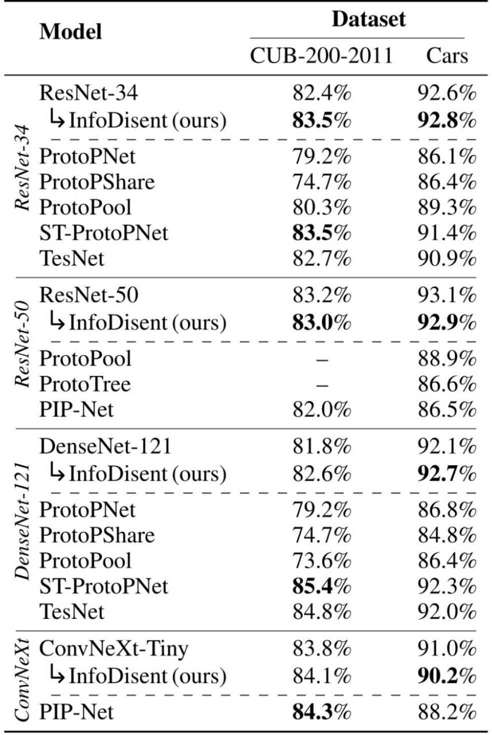

Information Bottleneck
Orthogonal Transform
max (ReLU( )) − max (ReLU(−)) =
Output Logits
Input
Frozen Backbone
Pooling Important Features
FC Layer with Positive Values

Figure 2: Our image classification interpretation model, InfoDisent, features three main components: a pre-trained backbone, a pooling layer for key features, and a fully connected layer. The CNN/transformer backbone, with frozen weights, is not further trained. The pooling layer extracts features from the last transformer or convolutional layer and identifies key positive and negative features. These are then combined into a dense vector, which is processed by a fully connected linear layer with nonnegative entries in the final stage.

To visualize the decision we show the patch on the input image which corresponds to the greatest positive pixel in the representation space.

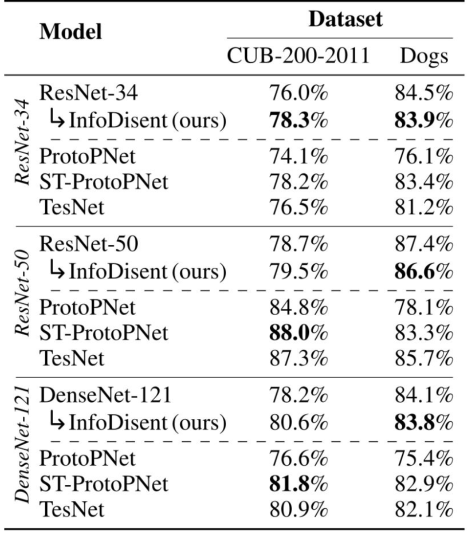


Results



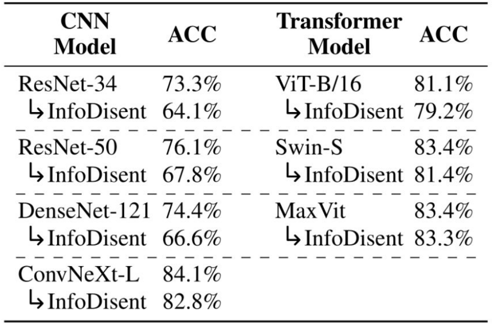
	Model	Dataset	
		CUB-200-2011	Cars
ResNet-34	ResNet-34	82.4%	92.6%
	↳InfoDisent (ours)	**83.5%**	**92.8%**
	ProtoPNet	79.2%	86.1%
	ProtoPShare	74.7%	86.4%
	ProtoPool	80.3%	89.3%
	ST-ProtoPNet	**83.5%**	91.4%
	TesNet	82.7%	90.9%
ResNet-50	ResNet-50	83.2%	93.1%
	↳InfoDisent (ours)	**83.0%**	**92.9%**
	ProtoPool	–	88.9%
	ProtoTree	–	86.6%
	PIP-Net	82.0%	86.5%
DenseNet-121	DenseNet-121	81.8%	92.1%
	↳InfoDisent (ours)	82.6%	**92.7%**
	ProtoPNet	79.2%	86.8%
	ProtoPShare	74.7%	84.8%
	ProtoPool	73.6%	86.4%
	ST-ProtoPNet	**85.4%**	92.3%
	TesNet	84.8%	92.0%
ConvNeXt	ConvNeXt-Tiny	83.8%	91.0%
	↳InfoDisent (ours)	84.1%	**90.2%**
	PIP-Net	**84.3%**	88.2%

Table 1: Accuracy comparison of interpretability models using standard CNN architectures (utilized in explainable models) trained on cropped bird images of CUB-200-2011, and Stanford Cars (Cars). Our approach demonstrates superior performance across nearly all the datasets and models considered. For each dataset and backbone, we boldface the best result in the class of interpretable models.



	Model	Dataset	
		CUB-200-2011	Dogs
ResNet-34	ResNet-34	76.0%	84.5%
	↳InfoDisent (ours)	**78.3%**	**83.9%**
	ProtoPNet	74.1%	76.1%
	ST-ProtoPNet	78.2%	83.4%
	TesNet	76.5%	81.2%
ResNet-50	ResNet-50	78.7%	87.4%
	↳InfoDisent (ours)	79.5%	**86.6%**
	ProtoPNet	84.8%	78.1%
	ST-ProtoPNet	**88.0%**	83.3%
	TesNet	87.3%	85.7%
DenseNet-121	DenseNet-121	78.2%	84.1%
	↳InfoDisent (ours)	80.6%	**83.8%**
	ProtoPNet	76.6%	75.4%
	ST-ProtoPNet	**81.8%**	82.9%
	TesNet	80.9%	82.1%

Table 2: Classification accuracy on full CUB-200-2011, and Stanford Dogs datasets by competing approaches using different CNN backbones. For each dataset and backbone, we boldface the best result in the class of interpretable models.



CNN Model	ACC	Transformer Model	ACC
ResNet-34	73.3%	ViT-B/16	81.1%
↳InfoDisent	64.1%	↳InfoDisent	79.2%
ResNet-50	76.1%	Swin-S	83.4%
↳InfoDisent	67.8%	↳InfoDisent	81.4%
DenseNet-121	74.4%	MaxVit	83.4%
↳InfoDisent	66.6%	↳InfoDisent	83.3%
ConvNeXt-L	84.1%		
↳InfoDisent	82.8%		

Table 3: Classification accuracy (ACC) on ImageNet dataset by competing approaches using different CNN backbones.

Thank you!