

Comparison of convolutional neural networks and transformer architectures for image classification using explanation methods

Mateusz Stączek

Supervisor

Elżbieta Sienkiewicz, PhD

Warsaw, 2.12.2024

Table of contents

- Introduction
- Contribution
- Methodology
- Results
- Summary

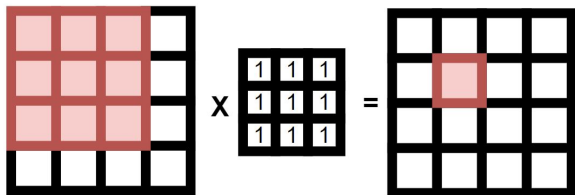


Introduction – Image classification

There are 2 popular architectures of neural networks for image classification:

CNNs - Convolutional Neural Networks

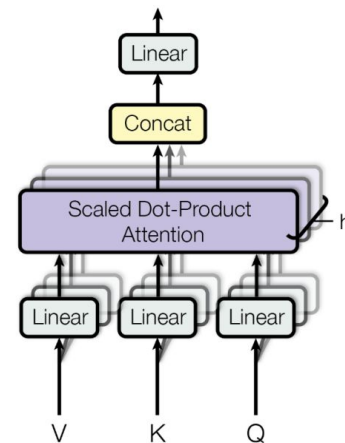
- Based on 2D convolutions



Transformers

- Based on attention mechanism
- Splits image into a grid of square patches

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$



Attention image source:

Vaswani et al., „Attention is All you Need“, in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017.

Introduction – Explanations

Image: original images and explanations highlighting influential regions.

Questions:

- Do the explanations change depending on model architecture (CNN/transformer)?
- What else affects the explanations?

Explanations of a CNN and a transformer using Grad-CAM

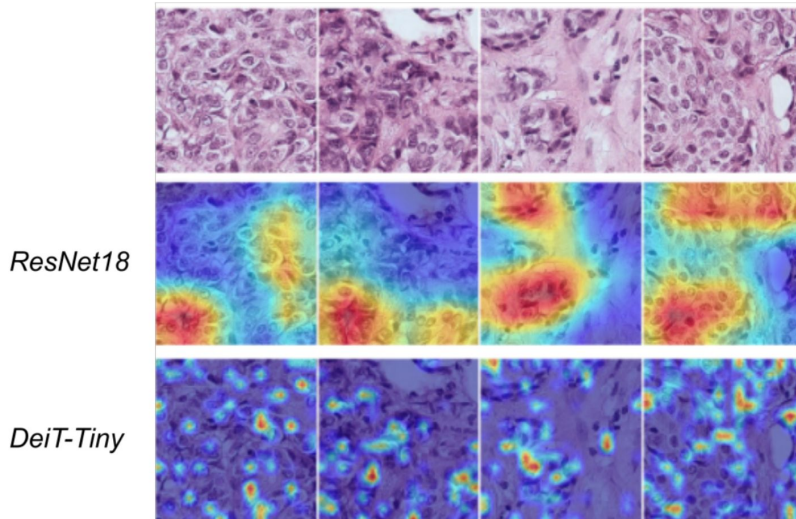


Image source:

L. Deininger, B. Stimpel, A. Yüce, et al., „A comparative study between vision transformers and CNNs in digital pathology”, ArXiv, 2022.

Contribution

1. **Quantitative comparison of 14 models** for image classification of 2 architectures:
 - a. 9 CNNs models,
 - b. 5 transformers.
2. Compare the results for **3 explanation methods**:
 - a. Including one model agnostic method, and 2 based on gradients,
 - b. Overall, compare over 10 000 explanations.
3. **Discover that the explanations differ in a way which does not align with model architectures**:
 - a. Explain the influence of their internal structure and how the explanation methods differ.
4. Publish the code on Github.

Methodology



Methodology – Dataset and models

Dataset: *Imagenette*

- Subset of the ***ImageNet***, popular benchmark dataset.



Imagenette source: <https://github.com/fastai/imagenette>

Models:

- 9 CNNs and 5 transformers.
- **Pretrained on *ImageNet*.**
- Almost all of similar size and **accuracy around 80%.**

Included models such as:

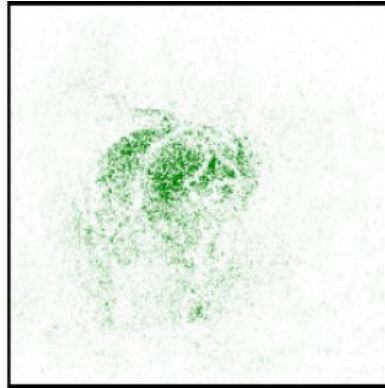
- ResNet, DenseNet, EfficientNet,
- ViT, DeiT, Swin, PVT.

Methodology – Explanation Methods

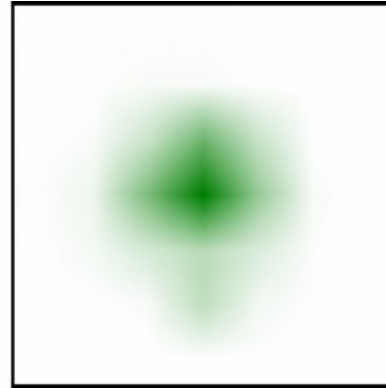
Original Image



Integrated
Gradients



Grad-CAM



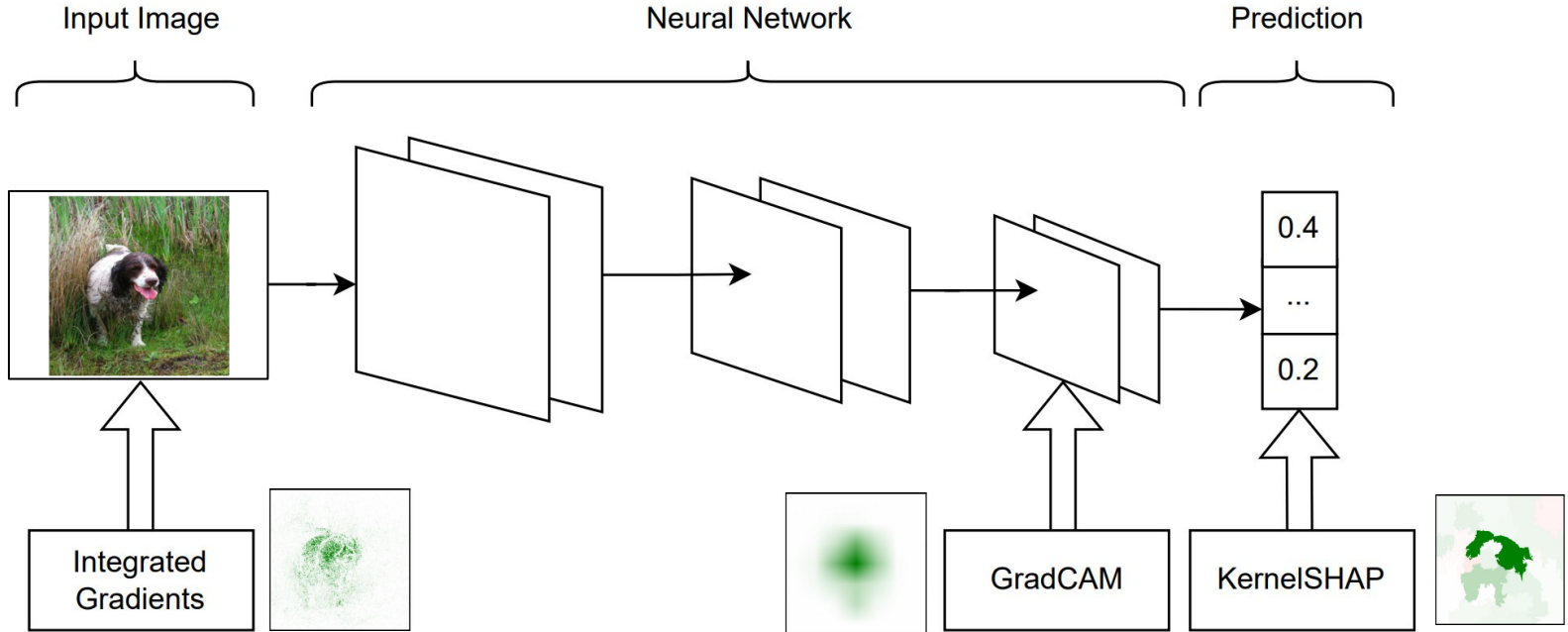
KernelSHAP



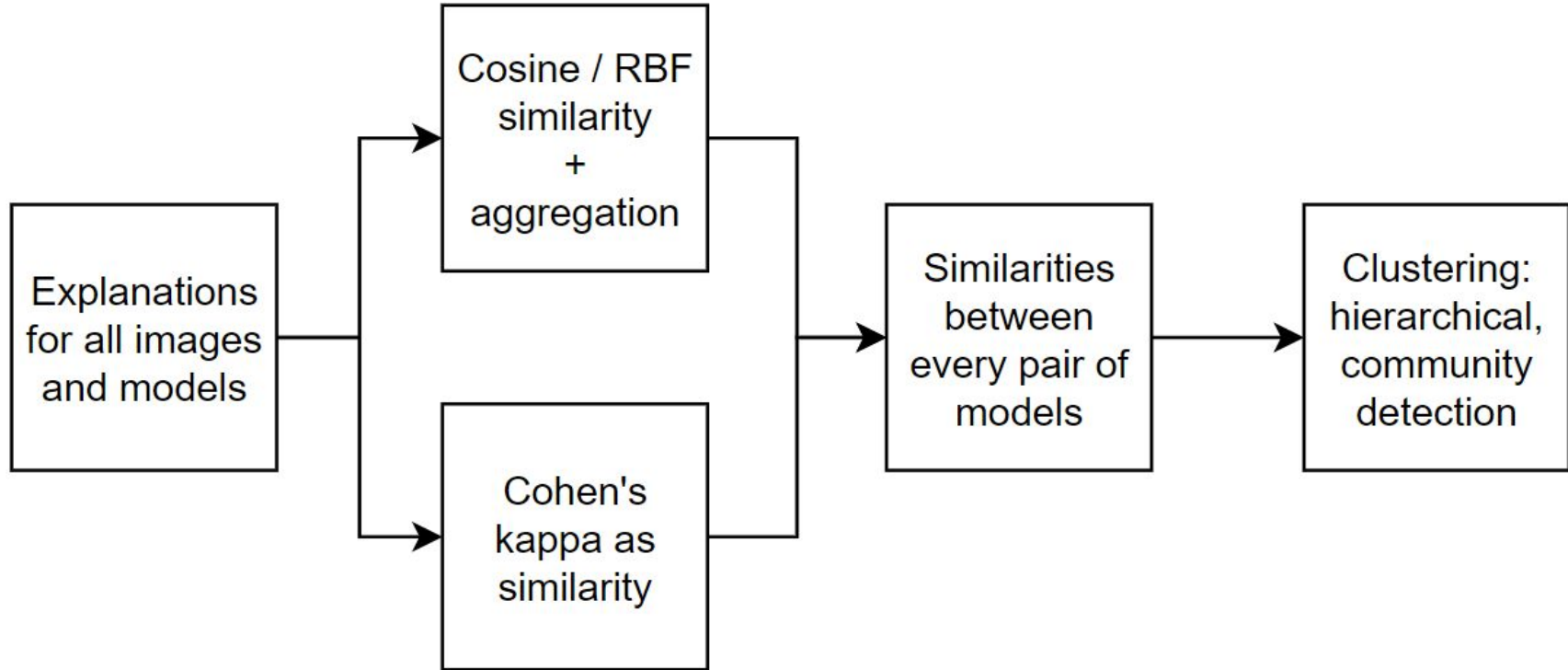
Explanations computed for a ResNet18 CNN.

Methodology – Explanation Methods

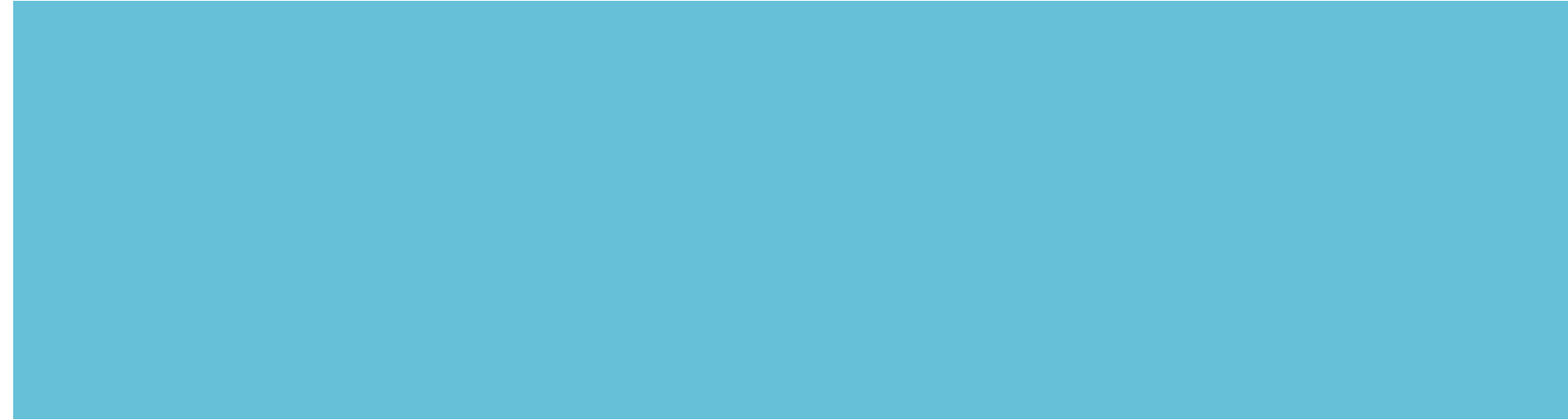
Places analyzed by different explanation methods



Methodology – Solution architecture



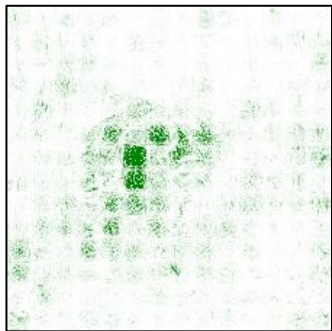
Results



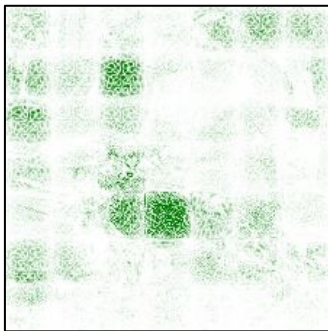
Results – Visualizations

In total, over 10 000 explanations were computed.

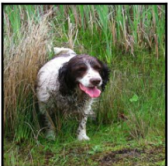
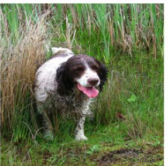

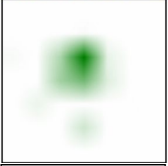
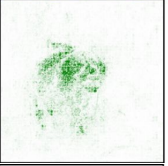
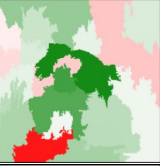
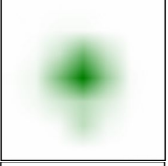
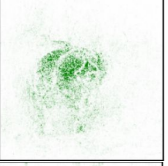
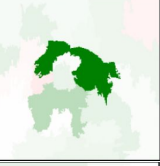
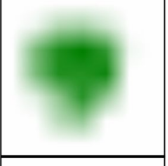
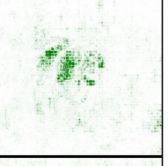

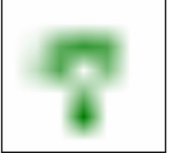
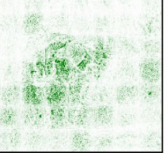
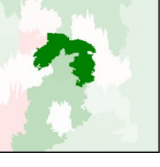
For uniform-scale transformers, Integrated Gradients shows a clear grid-like pattern:



DeiT



ViT

	Grad-CAM	Integrated Gradients	KernelSHAP
Original Image			
ConvNeXtV2 Nano CNN			
ResNet18 CNN			
Swin T transformer			
ViT B 32 transformer			

Results – Pairwise model dissimilarities

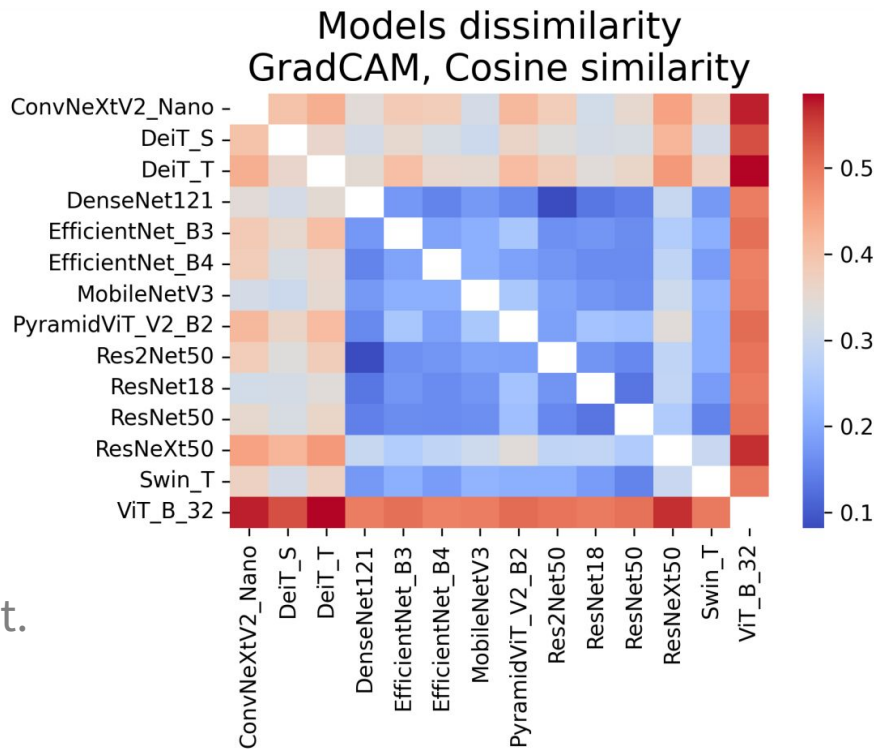
Dissimilarity for every pair of models, as a heatmap.

Blue:

- most CNNs are similar.

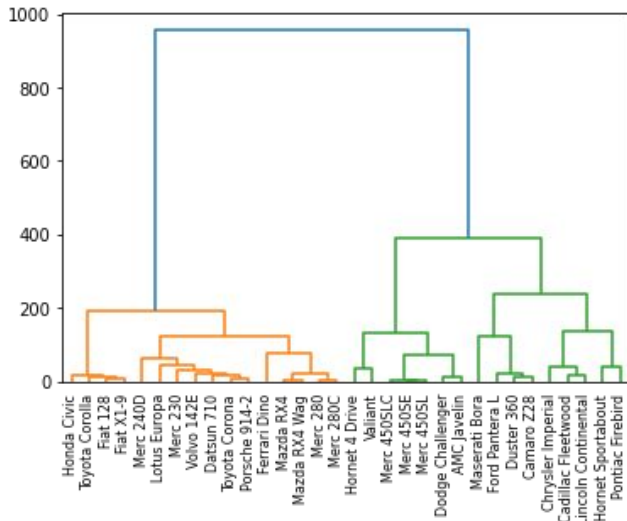
Red:

- ViT transformer differs the most.

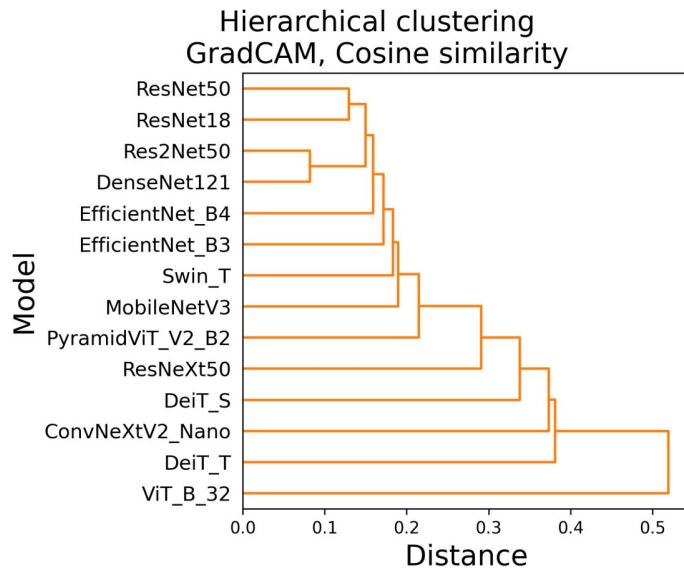


Dendrograms reminder

2 clusters are visible: orange and green.



Below, only 1 cluster gradually grows.



Source: <https://python-graph-gallery.com/dendrogram/>

Results – Clustering

Result: all models join a single cluster.

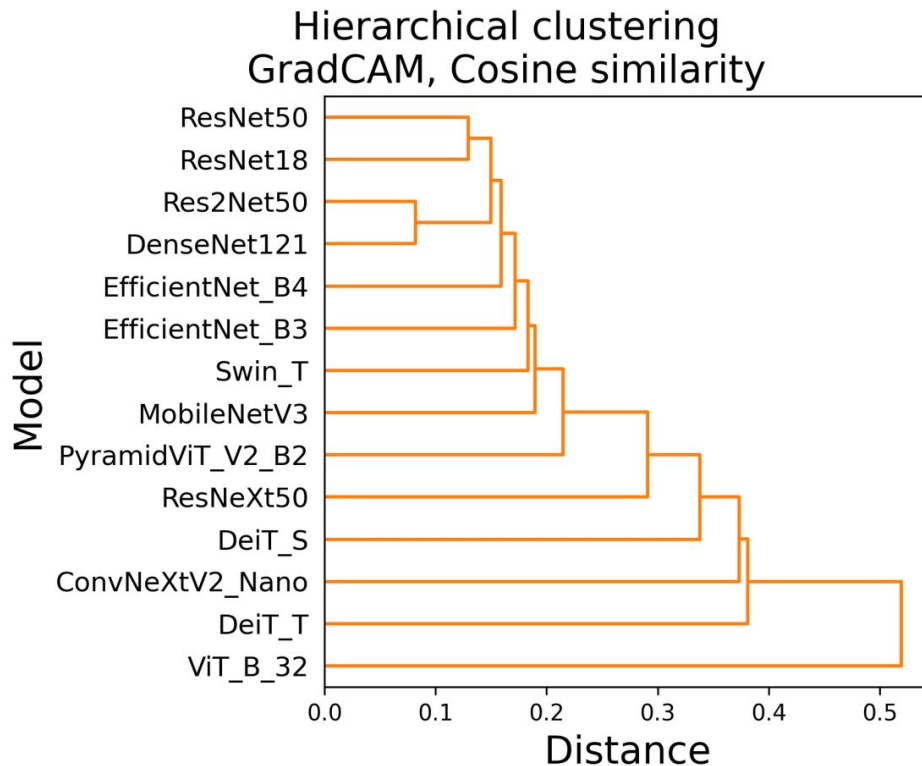
All dendrograms, based on:

- 2 similarity metrics,
- 3 explanation methods,

produced similar results:

- **no clear clusters were formed.**

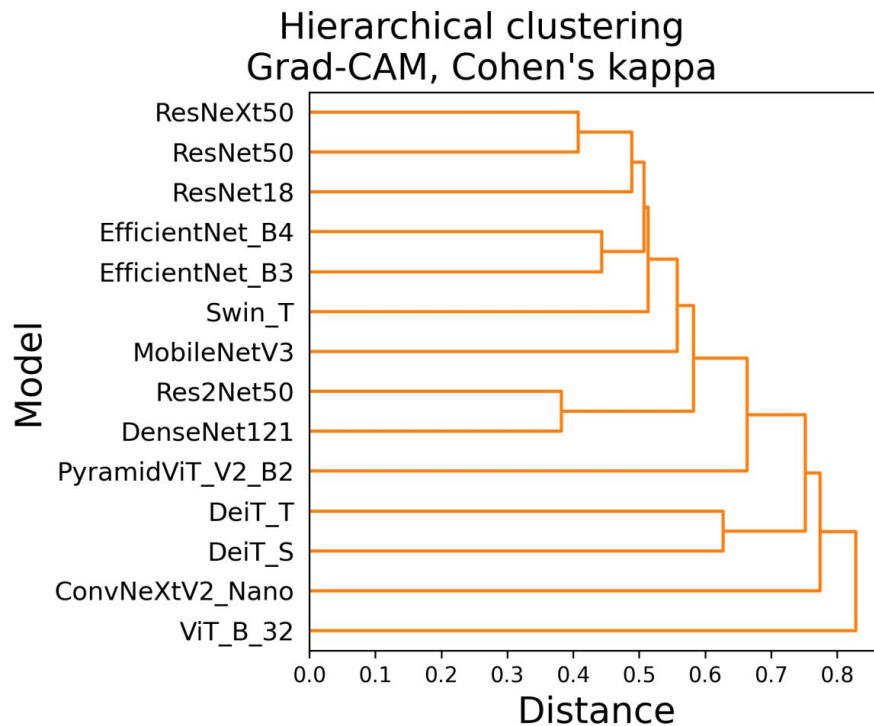
Community detection gave same conclusions.



Results – Cohen's kappa models agreement

Models are not grouped into visible separate clusters.

This confirms earlier results.

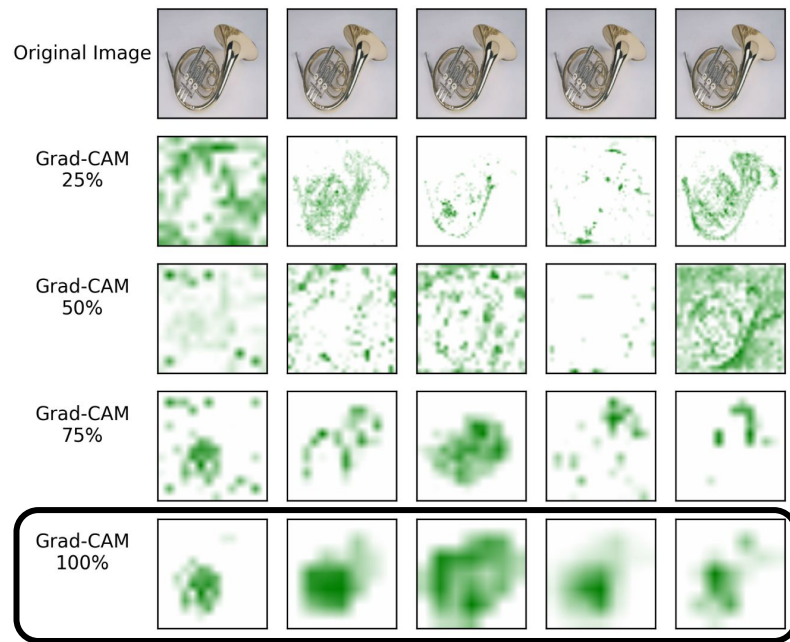


Results – Additional experiments

1. Computing Grad-CAM at different layers.
 - a. Last layer gave the best results.
2. Discover some bias in the training set.
 - a. Masking a fish did not have much influence.



Explanations at different parts of the models



Columns - different models

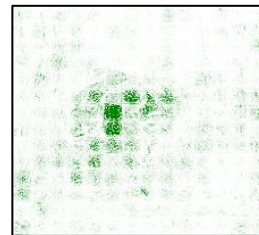
Summary



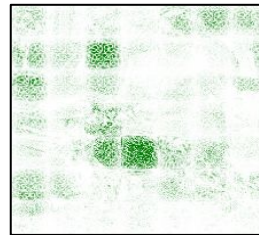
Conclusions

1. Surprisingly, results of clustering **do not align with model architectures**.
 - a. No clear clusters were formed during clustering – we cannot guess which model is what architecture.
2. Some explanation methods are negatively affected by model's internal structure.
 - a. **Integrated Gradients sometimes produces grid-like pattern.**
3. Resolution of Grad-CAM explanations depends on the size of the analyzed layer.
 - a. Usually, it decreases to just 7x7 pixels, but for some models it is higher (and constant), at 14 by 14 pixels.

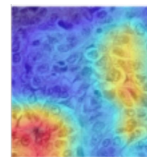
Integrated
Gradients



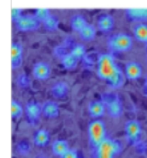
DeiT



ViT



ResNet18



DeiT-Tiny

Future work

1. Analyze a more diverse selection of models – for example **hybrid transformers**.
 - a. They use convolutional layers and attention mechanism.
2. Add **more explanation methods**.
 - a. Could bring new perspective when looking at the explanations.
 - b. Also, try different hyperparameters, for example analyze different layers with Grad-CAM.
3. Try different or more complex datasets.
 - a. **Medical images**, where **context matters** and a model has to **detect more than 1 element**.
4. Compare the explanations differently.
 - a. Other clustering methods.
 - b. Another approach.

Bibliography

1. **Similar comparison but of CNN models only:** X. Li, H. Xiong, S. Huang, S. Ji, and D. Dou, „Cross-model consensus of explanations and beyond for image classification models: an empirical study”, Machine Learning, vol. 112, pp. 1627–1662, 2021.
2. **Integrated Gradients:** M. Sundararajan, A. Taly, and Q. Yan, „Axiomatic Attribution for Deep Networks”, ser. ICML’17, 2017, pp. 3319–3328
3. **Grad-CAM:** R. R. Selvaraj et al., „Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”, in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.
4. **KernelSHAP:** S. M. Lundberg and S.-I. Lee, „A Unified Approach to Interpreting Model Predictions”, in Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.
5. **Models library:** R. Wightman, PyTorch Image Models, <https://github.com/rwightman/pytorch-image-models>, 2019.
6. **Dataset:** J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, „ImageNet: A large-scale hierarchical image database”, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

The end

Thank you
for your attention



The end

Thank you
for your attention

Additional / removed slides are below.

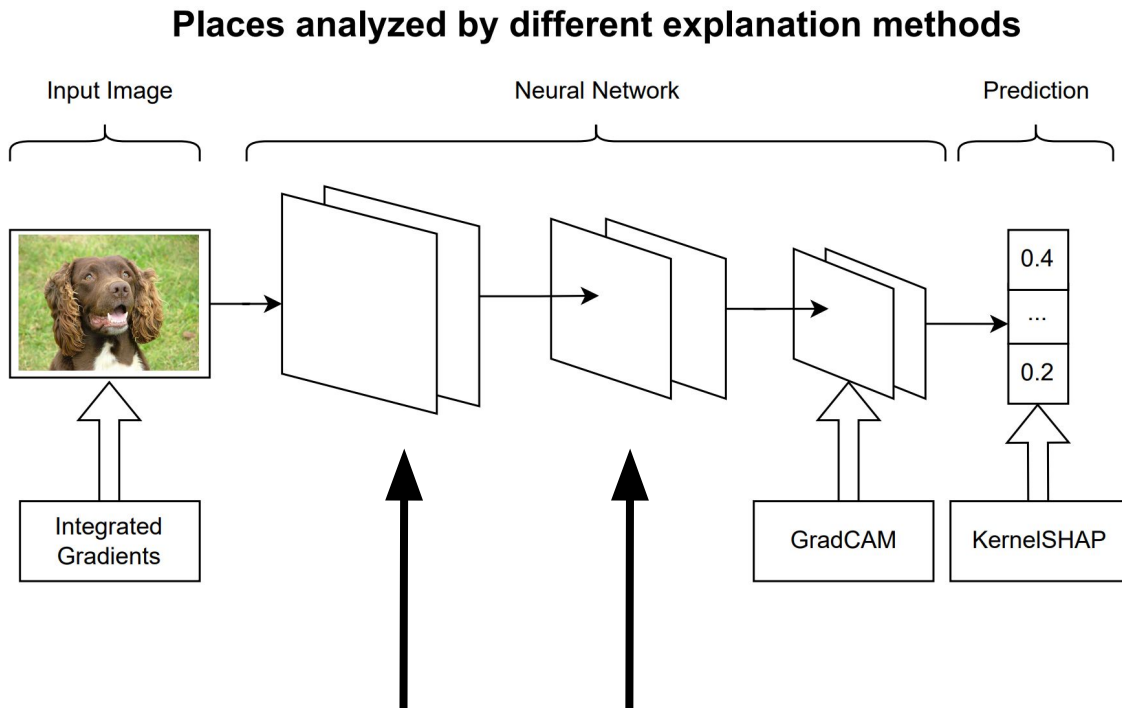
Results – Grad-CAM but inside

Typically:

- Grad-CAM explains one of the last layers

But:

- It can also explain the previous layers



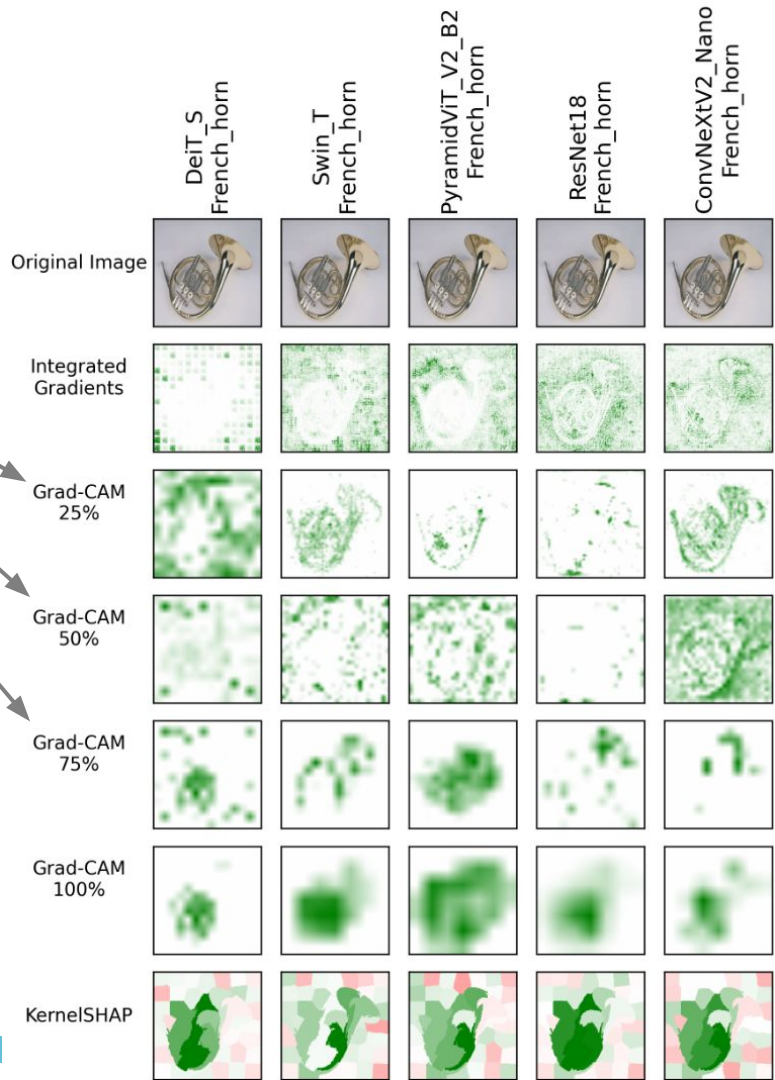
Results – Grad-CAM but inside

We computed a few Grad-CAM explanations also for:

- Layer after $\frac{1}{4}$ of all layers – called 25%
- Half of the layers – called 50%
- Layer after $\frac{3}{4}$ of all layers – called 75%

Results:

- New layers give less clear explanations.
- The last layers work best for Grad-CAM.



Results – Community detection

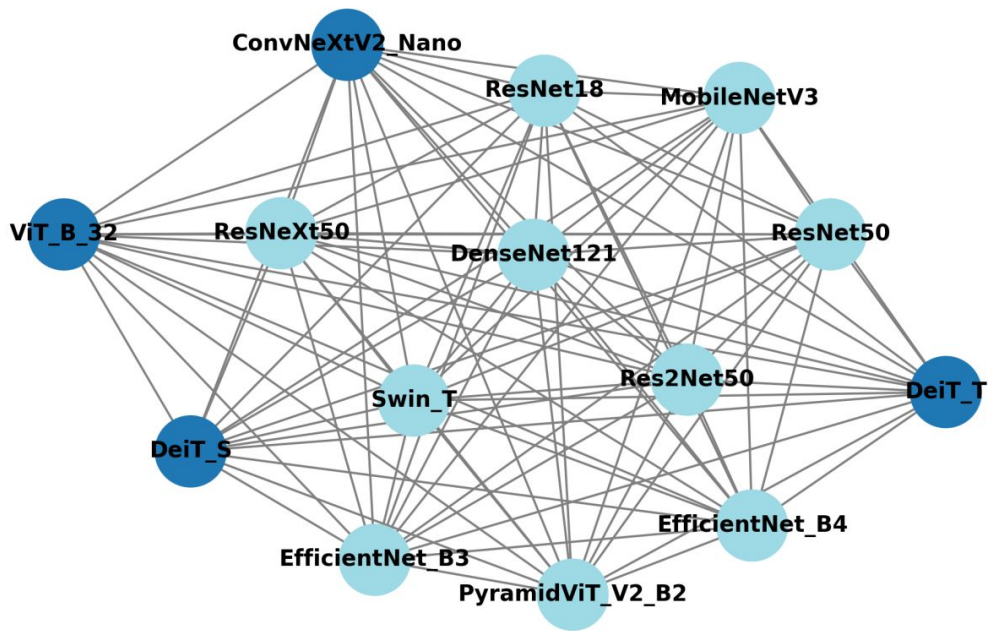
Community detection clustering:

- **Weighted graph,
by similarity of models (nodes)**

Result:

- Very similar to taking a slice from a dendrogram,
- *Only color of nodes matters.*

Graph community detection, param=1.06
GradCAM, Cosine similarity

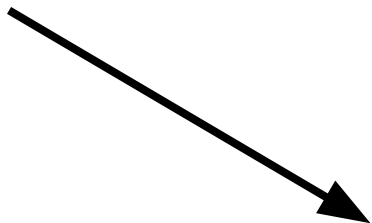


Appendix – Dataset bias

Top - original image, with a fish



Bottom - modified, without a fish



Result:

- All models classify the top image correctly.
- All models could still a fish in the bottom image.
 - It was a different species of fish though.
- Explanations often highlighted the person.



Methodology – Solution architecture 1

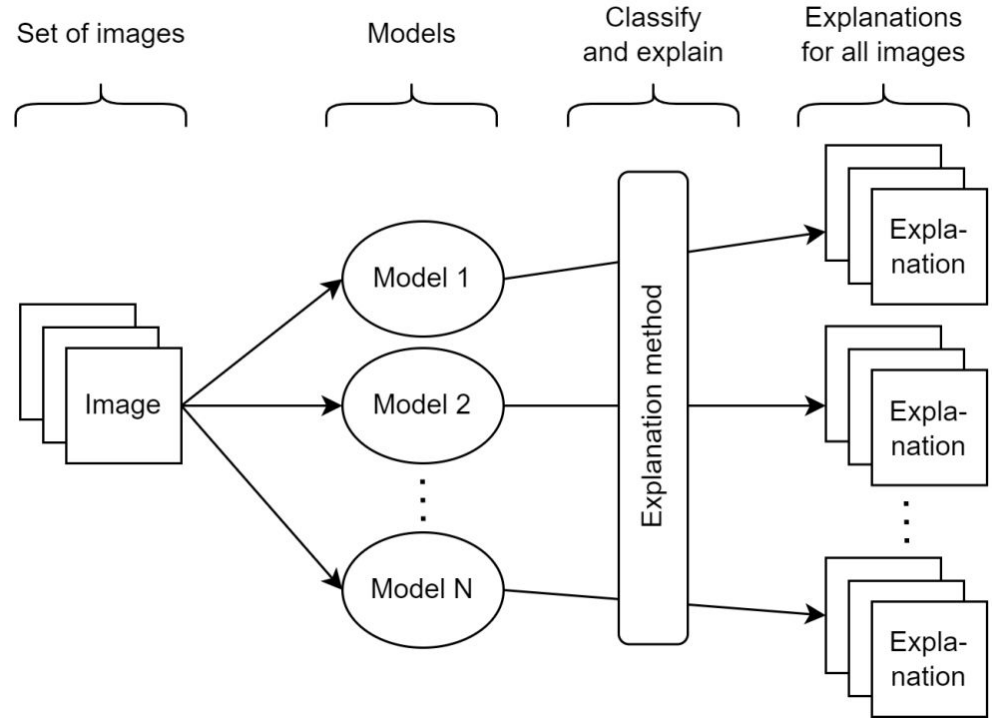
Steps:

1. Compute explanations

- a. 128 images
- b. 14 models
- c. 3 explanation methods

2. Compare pairs of explanations

- a. Only compare explanations of the same image
- b. Aggregate the results

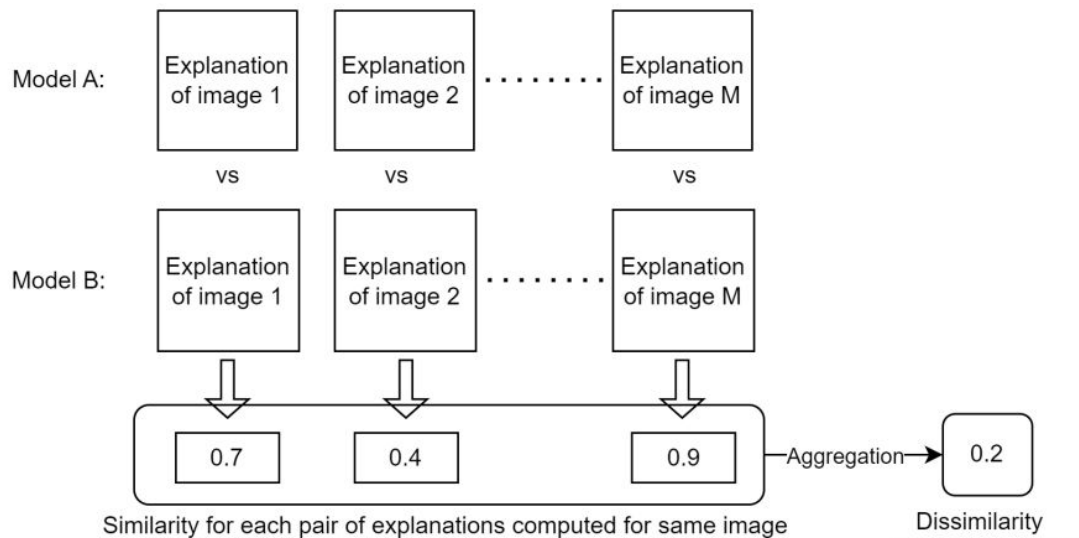


Methodology – Solution architecture 2

Steps:

1. Compute explanations
 - a. 128 images
 - b. 14 models
 - c. 3 explanation methods
2. **Compare pairs of explanations**
 - a. **Only compare explanations of the same image**
 - b. **Aggregate the results**

Similarity metrics: cosine or RBF
Alternative: Cohen's kappa coefficient



Methodology – Models

Models:

- 5 transformer models
- 9 CNN models

All of similar size and accuracy.

All pretrained on ImageNet.

Model Name	Architecture	#param (M)	Accuracy
ConvNeXt V2-N	CNN	16	81.9
DenseNet121	CNN	8	75.0
EfficientNet-B3	CNN	12	81.6
EfficientNet-B4	CNN	19	82.9
MobileNetV3	CNN	6	75.2
ResNet18	CNN	12	71.5
ResNet50	CNN	25	80.4
ResNeXt50	CNN	25	80.5
Res2Net50	CNN	25	78.0
DeiT-S	Transformer	22	79.8
DeiT-T	Transformer	6	72.2
PVTv2-B2	Transformer	25	82.0
Swin-T	Transformer	29	81.3
ViT-B/32	Transformer	88	81.3

Methodology – Cohen's kappa coefficient

Treat each explanation pixel as important (1) or not (0).

Measure agreement (similarity) between explanations of a pair of models:

p_o – fraction of agreement

p_e – fraction of disagreement

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

p_e – Subtracting fractions of disagreement accounts for agreement by chance.

Methodology – Similarity metrics

For **similarity of a pair of explanations**:

- RBF $\exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$

- Cosine similarity $\frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$

To **aggregate similarities for a pair of models**:

- Dissimilarity based on mean and variance of similarities

$$\text{dissimilarity}(a, b) = \sqrt{\left(1 - \frac{1}{N} \sum_{i=1}^N x_i\right)^2 + s_X^2}$$

Methodology – Similarity metrics

For a pair of explanations \mathbf{x} and \mathbf{y} for the same image and different models:

- RBF similarity $\exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$

- Cosine similarity $\frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$

For a pair of models - compute mean similarity of all explanations.

Methodology – Clustering

Hierarchical clustering – based on distances between items/clusters

- We use **agglomerative clustering**
- Linkage: average

- Distance between clusters:

$$d(A, B) = \frac{1}{|A| \cdot |B|} \sum_{i=1}^N \sum_{j=1}^M d(a_i, b_j)$$

Community detection – graph-based approach:

- Treat similarity of a pair of models as a weight of an edge
- Gives results similar to hierarchical clustering

Methodology – Dissimilarity metric

Distributions of similarities have different widths, so we used the following metric:

$$\text{dissimilarity}(a, b) = \sqrt{\left(1 - \frac{1}{N} \sum_{i=1}^N x_i\right)^2 + s_X^2}$$

a, b - models

x_i - similarity of explanations for models a and b of i -th image

N - number of explanations compared

First part – higher similarity = closer to 0 is better

Second part – lower variance of similarity = closer to 0 is better