IteROAR: Quantifying the Interpretation of Feature Importance Methods

Sebastian Palacio[1,2] **Federico Raue**[1] **Tushar Karayil**[1] **Jörn Hees**[1] **Andreas Dengel**[1,2]
[1] German Research Center for Artificial Intelligence (DFKI)
[2] TU Kaiserslautern
Kaiserslautern, Germany
`first.last@dfki.de`

Abstract

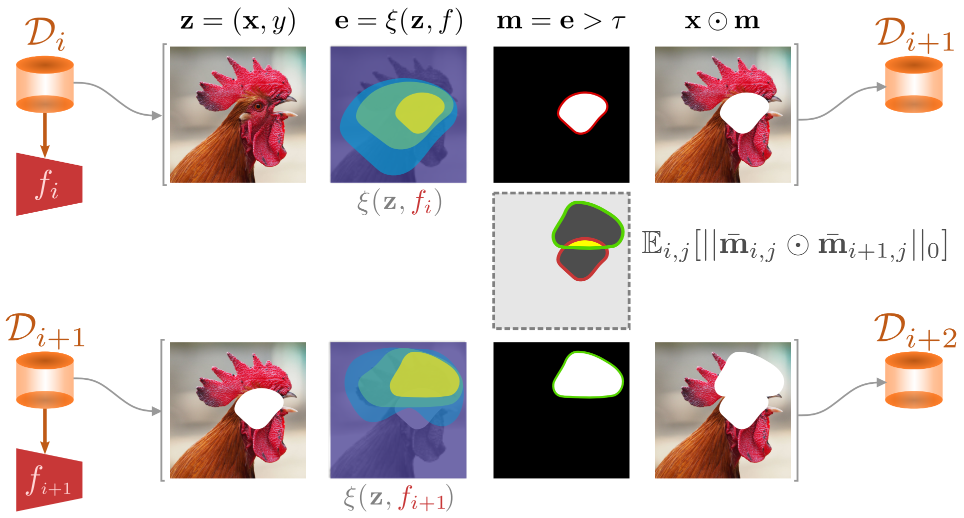
Feature importance methods are frequently used for XAI. However, the information they convey (i.e., their *interpretation*) has been established mostly through qualitative evaluations and, upon further inspection, their extent and usefulness is now being questioned. ROAR, a study by Hooker et. al. suggests that most methods fail at highlighting the most important parts of the image, namely those that should trigger a correct answer. We propose ***IteROAR***, an iterative extension to ROAR that allows a more comprehensive test of the consistency with which common attribution methods (e.g., CAM, GradCAM) focus on areas that contribute most to the model's prediction. With ***IteROAR*** a reference model is trained on images whose most salient features get progressively occluded. In contrast to ROAR, ***IteROAR*** evaluates feature importance maps each time an occlusion is made, which enables the quantification of additional properties of importance maps like the expected cumulative masked area or the mean mask overlap. These metrics constitute a more comprehensive and stricter evaluation of the properties of feature importance maps, and help elucidate the extent by which feature attribution maps convey the information that has been empirically associated to them.

1 Introduction

Feature importance methods are functions that rank which part of an input elicits a stronger response for a prediction model. In practice, these methods have been used as explanations for deep neural networks and image classification problems. Algorithms like Class Activation Mapping (CAM) [24] produce heatmaps whose interpretation has consolidated, albeit intuitively, as the area that corresponds to the predicted class. This interpretation is shared among numerous methods that build upon CAM [4, 17, 21, 18, 20]. However, some studies found that said interpretation has broad implications that cannot be guaranteed. Among them, attribution methods fail at providing insights regarding the features extracted by the model e.g., when the same area is highlighted for two different classes [1, 15].

More recently Hooker et al. [10] introduced ROAR; a benchmark showing that popular feature importance methods perform no better than a random baseline at ranking importance of the input features. This benchmark consists on measuring a model's performance by retraining it on samples after an increasing percentage of their input has been occluded, according to an initial estimation of feature importance. The pillar of said benchmark is that progressive occlusion of features (as ranked by the feature importance method) should yield a monotonic decrease in performance.

We argue that this assumption comes with critical caveats that said benchmark does not capture. For example, a class can often be defined by a non-overlapping set of features (e.g., a rooster can be identified by detecting either their beak, or their crest). A model that has a strong response to one kind



$\mathcal{D}_i$
$\mathbf{z} = (\mathbf{x}, y)$
$\mathbf{e} = \xi(\mathbf{z}, f)$
$\mathbf{m} = \mathbf{e} > \tau$
$\mathbf{x} \odot \mathbf{m}$
$\mathcal{D}_{i+1}$
f_i
$\xi(\mathbf{z}, f_i)$
$\mathbb{E}_{i,j}[||\bar{\mathbf{m}}_{i,j} \odot \bar{\mathbf{m}}_{i+1,j}||_0]$
$\mathcal{D}_{i+1}$
$\mathcal{D}_{i+2}$
f_{i+1}
$\xi(\mathbf{z}, f_{i+1})$

Figure 1: Overview of ***IteROAR***. A dataset $\mathcal{D}_i$ is used to train a model f_i with which a feature importance method ξ outputs a map that can be thresholded to create a binary mask $\mathbf{m}$. This mask is used to create a modified version of the dataset $\mathcal{D}_{i+1}$ by multiplying it with the original input. This process is then repeated I times. The sequence of masks measures if ξ focuses on areas that progressively affect performance, or areas that have already been occluded (i.e., have no information).

of feature does not imply that its absence makes the other features less discriminating. Therefore, estimating feature importance based only on the original sample (as done in ROAR), does not capture the shift in emerging features that gain importance in the absence of others.

Moreover, an unintended property of learning-based systems is that they often excel by relying on shortcuts that are particular to the benchmark, while not necessarily reflecting the general features of the task [7]. Said shortcuts materialize in ML models as sets of non-robust features [11] which lead to unstable predictions (i.e., the output changes drastically with a small modification to the input). Quantifying importance based on a single estimation limits the kind of features that get highlighted and hence, those more critical for the evaluation.

We introduce **Ite**rative **ROAR** (***IteROAR***): a general benchmark inspired by ROAR that allows a thorough evaluation of feature importance methods in three fundamental ways:

- In the absence of some of the salient characteristics, ***IteROAR*** allows feature importance methods to shift focus to a different set of features each time that the model gets re-trained.
- ***IteROAR*** makes it possible to validate the ad-hoc interpretation given to feature importance methods regarding the faithfulness [2] with which said methods highlight the most class-relevant input features.
- ***IteROAR*** accounts for the class-relevant features that are both suppressed *and* introduced by the masking operation.

By gathering consecutive responses of the feature importance method, ***IteROAR*** gives a broader account of the features that the model responds to, be it of the robust or non-robust kind. Therefore, we say that ***IteROAR*** serves as an adequate estimation of *"Morgan's Canon for Machine Learning"* [7] i.e., whether a prediction is the result of high-level abilities or is it simply exploiting a shortcut.

2 Related work

Feature importance algorithms are popular explanation methods, as their output helps out in inferring otherwise opaque behaviors of deep neural networks. Although these techniques are not the only way of generating explanations, they are often preferred as they are convenient to visualize and interpret.

One of the first methods to estimate relevance of input features is called Class Activation Mapping (CAM) [24]. It is defined only for fully convolutional neural networks and works by constructing a coarse saliency map with the weighted average of the last convolutional activation. Later on, multiple methods addressed the limitations of CAM by using gradients [17, 4, 20, 21] or other intermediate activations [22, 3].

These methods or variations thereof, have been adopted as a way to justify predictions of black-box models used for high-stake scenarios [23, 12]. However, these same methods are known to be inconsistent with the expectations regarding their interpretation [15, 19].

The benchmark proposed by Hooker et al. [10] is a comprehensive mechanism to test one of the properties that feature attribution maps have, namely the relation between highly salient areas and their impact in a model's performance. We build ***IteROAR*** based on a similar premise, and extend the properties that can be evaluated by (1) testing whether attribution maps highlight areas whose information has been occluded, (2) how well is the information being occluded by the masks, and (3) if progressive occlusions lead to monotonic affectations of the model's accuracy.

3 Methodology

In a nutshell, ***IteROAR*** works as follows: given a dataset $\mathcal{D}_i = \{\mathbf{z}_{i,j}\}, \mathbf{z}_{i,j} = (\mathbf{x}_{i,j}, y_j), j \in \{1, \ldots, N\}$, a prediction model f_i is trained on $\mathcal{D}_i$, and a feature importance estimation method $\mathbf{e}_{i,j} = \xi(\mathbf{z}_{i,j}, f_i)$ computes the feature importance map for all samples in $\mathcal{D}_i$ w.r.t. the model f_i. A binary mask $\mathbf{m}_{i,j}$ is created from the resulting map $\mathbf{e}_{i,j}$ by applying a threshold τ such that individual values m_h of $\mathbf{m}_{i,j}$, $h = \{0, \ldots, A\}$ equal zero if $\mathbf{e}_{i,j} > \tau$ and 1 otherwise. Finally, a new dataset $\mathcal{D}_{i+1} = \{\mathbf{z}_{i+1,j}\}, \mathbf{z}_{i+1,j} = (\mathbf{x}_{i+1,j}, y_j)$ is created with the masked inputs $\mathbf{x}_{i+1,j} = \mathbf{x}_{i,j} \odot \mathbf{m}_{i,j}$ where $\odot$ corresponds to the Hadamard product. This entire process repeats I times resulting in a set of datasets $\{\mathcal{D}_0, \ldots, \mathcal{D}_I\}$. A graphical overview of ***IteROAR*** is shown in Figure 1.

With ***IteROAR*** we can test for several expected behaviours of feature importance methods (non-functional requirements of explanation methods as defined by Palacio et al. [14]). One such requirement is the aforementioned existence of different class-specific features that may gain relevance as other features get occluded. Another interesting property lies on the expected saliency of consecutive iterations: we can reasonably expect that computing the feature importance on samples whose previously salient areas have been masked, results in a map that highlights a different area outside of the mask. In other words, the overlap between masks from consecutive runs $\mathbf{m}_{i,j}, \mathbf{m}_{i+1,j}$ should be close to zero. In fact, unless the shape of the mask causes information leakage (e.g., by consistently masking the outline of an object), the expected overlap over all iterations should be close to zero. Formally, let $\bar{\mathbf{m}} = 1 - \mathbf{m}$ and $||\cdot||_0$ be the zero "norm", then the mean mask overlap (MMO)

$$\text{MMO} = \mathbb{E}_{i,j}[||\bar{\mathbf{m}}_{i,j} \odot \bar{\mathbf{m}}_{i+1,j}||_0] \tag{1}$$

quantifies how much of the feature importance lies on areas that have been occluded on the last iteration.

As masks are generated sequentially, a stronger constraint is that the latest mask cannot overlap with the area that has been masked so far. We call this the Mean Accumulated Mask Overlap (MAMO),

$$\text{MAMO} = \mathbb{E}_{i,j}\left[\left|\left|\left(\sum_{\iota=0}^{i} \bar{\mathbf{m}}_{\iota,j}\right) \odot \bar{\mathbf{m}}_{i+1,j}\right|\right|_0\right] \tag{2}$$

and it indicates how much of the total occluded area so far is still marked as highly relevant for the prediction. According to the intuitive interpretation of feature importance maps, we expect MAMO ≈ 0. Naturally, this property should hold as long as the value of τ guarantees that there are still enough class-relevant features for ξ to focus on. With this in mind, we constrain the values for τ and I such that the number of masked pixels after I iterations equals *at most* a portion of the image's area A. In other words, $||\bar{\mathbf{m}}_{0,j}||_0 \times I = \rho A$, $0 < \rho < 1$.

An even stronger expectation with respect to the masked area is that none of the masks should have any overlap:

$$\mathbb{E}_{i,j}[||\bar{\mathbf{m}}_{i,j} \odot \bar{\mathbf{m}}_{n,j}||_0] \approx 0 \quad i, n \in \{0, \ldots, I\}, i \neq n \tag{3}$$

A corollary of Equation 3 is that the cumulative masked area for the j-th sample should approximate the maximum allowed area ρA (as defined by the constraint above) after I iterations:

$$\text{CMA}_j = \left\| \sum_{i=0}^{I} \bar{\mathbf{m}}_{i,j} \right\|_0 \approx \rho A \tag{4}$$

We consider Equation 4 to hold as long as the model's accuracy stays above random chance (and no data leakage is observed). Therefore, we also monitor the model's test accuracy as masks get computed on each iteration.

4 Experiments

4.1 Setup

For ***IteROAR*** we adhere, for the most part, to the experimental setup outlined in Hooker et al. [10]. We use ResNet-50 as reference architecture for the classification model with He initialization [9]. Each combination of (dataset $\times$ feature importance method $\times$ masking operation) is repeated at least three times[1]. The constraint ρ that determines the maximum area that can get masked is set to 0.5 and following the setup by Chen et al. [5] we set the masking threshold τ to the 95th percentile.

4.2 Training and hyper-parameter tuning

As described in section 1, we start by loading the original dataset, training a ResNet-50 from scratch using data augmentation, a learning rate schedule and mini-batch SGD as the optimizer. The particulars of the hyperparameter space (optimizer, data augmentation, regularization, learning rate schedule) is determined by tracking the most accurate model on a validation set. During validation, the training set is split into 85%/15% sets and the latter is used for estimating validation metrics (cross-entropy loss and accuracy). Note that, in contrast to ROAR, we do not apply test time pre-processing [8] because of the required large computational budget, and because it does not have a direct impact on the metrics of interest for this work.

4.3 Feature importance estimator

For each iteration i, once the model has been trained to convergence using $\mathcal{D}_i$, we apply the feature importance estimator $\xi(\cdot,\cdot)$ for all samples in the test split of the corresponding dataset. For each feature importance map, the threshold τ to generate the binary mask corresponds to the 95th percentile of the values in the map. In order to favor comparable results, we analyze the feature importance methods in Hooker et al. [10]: Sensitivity Heatmaps, Guided Backprop, Integrated Gradients, SmoothGrad, SmoothGrad^2 and VarGrad. Moreover, we add the following commonly used feature importance methods to the test bench: Class Activation Mapping (CAM), GradCam and Occlussion Sensitivity.

A fundamental difference between ***IteROAR*** and ROAR is how progressive occlusions are generated. For ROAR the feature importance estimator $\xi(\cdot,\cdot)$ is computed only once on the original, non-occluded images, and progressive occlusions are generated by lowering the threshold τ each time. In ***IteROAR***, we compute the feature importance map anew for each iteration of the dataset $\mathcal{D}_i$, while leaving the threshold τ constant. This change alone allows us to test strong assumptions regarding the interpretation that is usually given to such feature attribution maps: the highlighted areas contain features that caused the model to issue a certain prediction. This implies that computing $\xi(\cdot,\cdot)$ on partially occluded samples, based on a network that has been trained on such samples, should not yield feature importance maps whose highlighted areas fall into previously occluded parts of the sample.

4.4 Masking

Next, the mask is used to occlude the thresholded area on the corresponding input sample. We propose two ways to fill in the occluded values: (1) with zeros, (2) with the mean value of the occluded pixels.

[1]depending on the availability of computational resources, we will run more trials.

4.5 Metrics

We report the distribution of the test accuracy over the number of iterations $i = 0, \ldots, I$. Moreover, the mean accumulated mask overlap MAMO and the distribution of mean cumulative masked area $\mathbb{E}[CMA_j]$ will be reported for each combination of (dataset $\times$ feature importance method $\times$ masking operation).

4.6 Datasets

We choose a selection of image datasets covering a wide range of spatial resolutions: CIFAR10 [13], STL-10 [6], and Imagenet [16]. This helps us verify how sensitive (if at all) feature importance methods are to the original spatial resolution. Note that, in order to evaluate all datasets using the same network architecture, we resize all images to 224$\times$224. Maintaining the original sizes of CIFAR10 and STL-10 would require a modification on the original ResNet-50 architecture, which would make these results less comparable across datasets.

4.7 Addressing data leakage through the masks

Even though the purpose of the masks is to occlude information on the original input, it is possible that the mask itself provides (or leaks) information about the class e.g., if the masks for one class are of a unique shape or occur in consistent locations.

Anticipating and measuring all possible features that masks may introduce quickly becomes untraceable. However, we propose a simple proxy metric to analyze the net effect of data leakage through masks: given a dataset $\mathcal{D}_i$, we compute the masks of the training and the test set, so that a modified version of ResNet-50 can be trained directly on the masks (the labels of the corresponding samples are preserved as labels for the masks). The intuition is that the classifier would do no better than random chance if the masks do not leak any class-relevant information. We propose running this experiment at least three times and record the test accuracy and loss along with the standard deviation. We then compare these results to a random baseline where labels are assigned at random to all masks.

Masks can leak information through position, size or shape. Altering size and shape can be addressed by applying morphological operations (e.g., dilation/erosion), randomized noise or by replacing them with their convex hulls. The mask's position (assuming the area is contiguous) can be altered via rotation, scaling, and transposition. In case the masks do cause data leakage (i.e., performance of ResNet-50 when trained on masks is significantly higher than random chance), we propose modifying the masks with the aforementioned operations, for each iteration of MAMO.

Acknowledgments and Disclosure of Funding

This work has been funded by BMBF the project ExplAINN (01IS19074)

References

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9525–9536, 2018.

[2] D. Alvarez-Melis and T. S. Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7786–7795, 2018.

[3] A. Binder, S. Bach, G. Montavon, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, pages 913–922. Springer, 2016.

[4] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[5] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: Deep learning for interpretable image recognition. *Advances in Neural Information Processing Systems*, 32: 8930–8941, 2019.

[6] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[7] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[8] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[10] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. *Advances in Neural Information Processing Systems*, 32:9737–9748, 2019.

[11] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Mądry. Adversarial examples are not bugs, they are features. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 125–136, 2019.

[12] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.

[13] A. Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[14] S. Palacio, A. Lucieri, M. Munir, S. Ahmed, J. Hees, and A. Dengel. XAI Handbook: Towards a unified framework for Explainable AI. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3766–3775, October 2021.

[15] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[18] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[19] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[20] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[21] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[22] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision(ECCV)*, 2016.

[23] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6428–6436, 2017.

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.