CAM-Based Methods Can See through Walls (Extended Abstract)[*]

Magamed Taimeskhanov[1] , **Ronan Sicre**[2] , **Damien Garreau**[1]
[1]Julius-Maximilians Universität Würzburg, CAIDAS, Würzburg, Germany
[2]Centrale Méditerranée, Aix-Marseille Univ., CNRS, LIS, Marseille, France
magamed.taimeskhanov@uni-wuerzburg.de, ronan.sicre@lis-lab.fr, damien.garreau@uni-wuerzburg.de

Abstract

CAM-based methods are widely-used post-hoc interpretability methods that produce a saliency map to explain the decision of an image classification model. The saliency map highlights the important areas of the image relevant to the prediction. In this paper, we show that most of these methods can incorrectly attribute an important score to parts of the image that the model cannot see. We show that this phenomenon occurs both theoretically and experimentally. On the theory side, we analyze the behavior of GradCAM on a simple masked CNN model at initialization. Experimentally, we train a VGG-like model constrained to not use the lower part of the image and nevertheless observe positive scores in the unseen part of the image. This behavior is evaluated quantitatively on two new datasets. We believe that this is problematic, potentially leading to mis-interpretation of the model's behavior.

1 Introduction

The recent advances of machine learning pervade all applications, including the most critical. However, deep learning models intrinsically possess many parameters, have complicated architectures, and rely on many non-linear operations, preventing the users to get a good grasp of the rationale behind particular decisions. These models are often called "black boxes" for these reasons [Benítez *et al.*, 1997]. In this respect, there is a growing need for interpretability of the models that are used, which gave birth to the field of eXplainable AI (XAI). When the model to explain is already trained, our main topic of interest, this is often called post-hoc interpretability [Lipton, 2018; Zhang *et al.*, 2021; Linardatos *et al.*, 2021].

In the specific case of image classification, the explanations provided to the user often take the form of a saliency map superimposed to the original image, for instance simply looking at the gradient with respect to the input of the network [Simonyan *et al.*, 2013]. The message is simple: the areas highlighted by the saliency maps are used by the network for the prediction. When the first layers of the network are convolutional layers [Fukushima, 1980], one can take advantage of this and look at the activations of the filters corresponding to the class prediction that we are trying to explain. Indeed, these first layers act like a bank of filters on the input image, and the degree to which they are activated gives us information on the behavior of the network. Thus the first layers possess a certain degree of interpretability, even though it can be challenging to aggregate the information coming from different filters. In any case, the next layers generally consist in a fully-connected neural network, thus suffering from the same caveats as other models. In addition, this second part of the network is equally important for the prediction, but is not taken into account in the explanations we provide if we simply look at activation values.

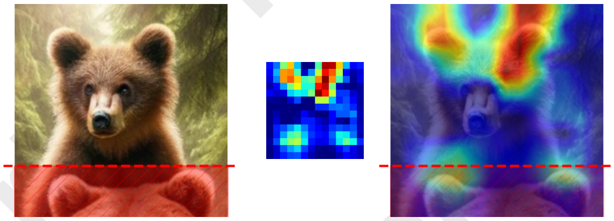

Figure 1: Example of GradCAM failure on a VGG-like model trained on the ImageNet dataset (masked $[\mathbf{VGG}]$, see Figure 4). *Left:* original image; *Middle:* GradCAM explanation before upsampling; *Right:* original image with GradCAM explanation overlayed as a heatmap. The network does not have access to the red part of the image, **but GradCAM does highlight some pixels in this area.**

To solve this problem, a natural idea is to weight each activation map depending on how the second part of the network uses it. In the case of a single additional layer, this is called *class activation maps* (CAM) [Zhou *et al.*, 2016], in which each activation map is weighted by its corresponding parameter in the output layer. The methodology was quickly generalized by [Selvaraju *et al.*, 2017], using the *average gradient* values of the subsequent layers instead, giving rise to GradCAM, arguably one of the most popular posthoc interpretability method for CNNs. Many extensions were proposed in the following years, we refer to [Zhang *et al.*, 2023] for a recent survey. Without being too technical, for all these methods,

[*] This is an extended abstract of [Taimeskhanov *et al.*, 2024]

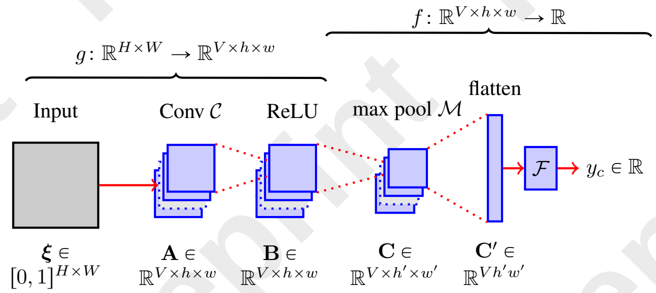
$f\colon \mathbb{R}^{V\times h\times w} \to \mathbb{R}$
$g\colon \mathbb{R}^{H\times W} \to \mathbb{R}^{V\times h\times w}$
Input
Conv $\mathcal{C}$
ReLU
max pool $\mathcal{M}$
flatten
$\mathcal{F}$
$y_c \in \mathbb{R}$
$\boldsymbol{\xi} \in [0,1]^{H\times W}$
$\mathbf{A} \in \mathbb{R}^{V\times h\times w}$
$\mathbf{B} \in \mathbb{R}^{V\times h\times w}$
$\mathbf{C} \in \mathbb{R}^{V\times h'\times w'}$
$\mathbf{C}' \in \mathbb{R}^{Vh'w'}$

Figure 2: The model used for the derivation of feature importance scores, [**CNN**]. The number of filters in the convolutional layer $\mathcal{C}$ is $V \in \mathbb{N}^\star$. The size of the max pooling filters $k' \in \mathbb{N}^\star$ is implicitly defined such that $(h', w') = \frac{1}{k'}(h, w)$ in $\mathbb{N}^\star$. The fully-connected neural network $\mathcal{F}$ takes $\mathbf{C}'$ as input and processes it through L layers with ReLU activation functions to produce a raw score y_c, without converting this score into a "probability."

the explanations provided consist in a weighted average of the activation maps.

A close inspection of each of these methods reveals that the coefficient associated to each individual map is global, in the sense that the same coefficient is applied to the whole map. The main message of this paper is that this can be problematic, since different parts of the activation map may be used differently by the subsequent layers. Worse, **some parts may even be unused by the subsequent network and still highlighted in the final explanation** (see Figure 1). Thus we believe that, while giving apparently more-than-satisfying results in practice, CAM-based methods should be used with caution, keeping in mind that some parts of the image may be highlighted whereas they are not even seen by the network.

This paper is inspired by a line of recent works concerned with the reliability of saliency maps claiming that solely relying on the visual explanation provided by a saliency map can be misleading [Kindermans *et al.*, 2019; Ghorbani *et al.*, 2019]. It is important to note that neither of these studies specifically challenges the reliability of CAM-based methods. This perspective on saliency maps is supported by the work of [Adebayo *et al.*, 2018], which introduces a randomization-based sanity check indicating that some existing saliency methods are independent of both the model and the data. We note that GradCAM passes the sanity checks proposed by [Adebayo *et al.*, 2018].

Draelos and Carin [2021], proposing HiResCAM, are less positive regarding GradCAM pointing out, as we do, that the use of a global coefficient can produce positive explanations where there should not be. Compared to our work, they provide few theoretical explanations. Posthoc interpretability methods in the image realm (not specific to CNN architectures) have been investigated by other works such as [Garreau and Mardaoui, 2021] which looked into LIME for images [Ribeiro *et al.*, 2016].

Taking another angle, [Heo *et al.*, 2019] directly attacks the

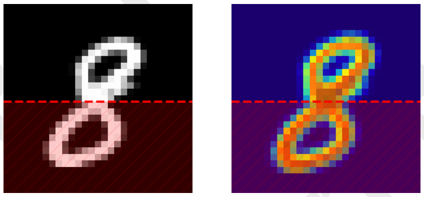

Figure 3: Illustration of Theorem 1 on an MNIST [LeCun *et al.*, 1998] digit (*left panel*). We set to zero the lower part of $\mathbf{W}$ for [**CNN**], initialize the filter values and remaining weights to i.i.d. $\mathcal{N}(0,1)$, and run GradCAM to get a saliency map (*right panel*). Even though our network does not see the red part of the image, **GradCAM does highlight some pixels in this area**, as predicted by Theorem 1.

reliability of GradCAM saliency maps by adversarial model manipulation, *i.e.*, fine-tuning a model with the purpose of making GradCAM saliency maps unreliable. This is achieved by using a specific loss function tailored to this effect. Our approach is different, as we simply force a strong form of sparsity in the model's parameters, not targeting a specific interpretability method.

In this paper, we start by looking at GradCAM theoretical behavior in Section 2. For a given simple CNN architecture described in Figure 2, we derive closed-form expressions for its explanations. Leveraging these expressions, we prove that GradCAM explanations are positive at initialization, even though a large part of the weights are set to zero. In Section 3, we demonstrate experimentally that this phenomenon remains true after training. To this extent, we proceed in two steps. First, we train to a reasonable accuracy a VGG-like model on ImageNet [Deng *et al.*, 2009] **which does not see the lower part of input images**. Then, we create two datasets consisting in superposition of images of the same class. We show experimentally that **CAM-based methods applied to this model wrongly highlights a large portion of the lower part of the images**, misleading the user by showing that the lower part is used for the prediction whereas, by construction it is not. Additionally, the code for all experiments is available online.[1] We conclude in Section 4.

2 Theoretical Results

Given the notations and [**CNN**] model introduced in Figure 2, in our notation:

Definition 2.1 (GradCAM)**.** For an input $\boldsymbol{\xi}$ and model [**CNN**], the GradCAM feature scores are given by

$$[\mathbf{GC}] := \sigma\left(\sum_{v=1}^{V} \alpha_v \mathbf{B}^{(v)}\right) \in \mathbb{R}_+^{h\times w},$$

where each $\alpha_v := \mathrm{GAP}\left(\nabla_{\mathbf{B}^{(v)}} f(\mathbf{B})\right) \in \mathbb{R}$. Here, GAP denotes the *global average pooling*, that is, the average of all values, and σ the ReLU as before.

Looking at Definition 2.1, whenever the underlying model is not too complicated, one can actually hope to derive a closed-

[1] https://github.com/MagamedT/cam-can-see-through-walls

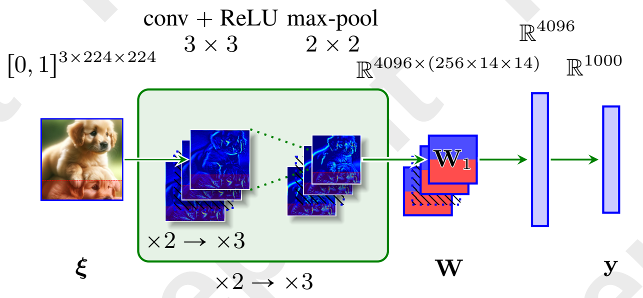

Figure 4: Our masked VGG16-based model trained on ImageNet with 87.0% top-5 accuracy. The down weights $\mathbf{W}_{:,\,:,\,-9:,\,:}$ are set to 0 and not updated during training. Only the up weights $\mathbf{W}_{:,\,:,\,:5,\,:}$ and the other parameters undergo training. This setting implies that every red part in the channels does not impact the prediction scores, meaning that they are not used. Symbol ×2 → ×3 means the model first uses the green block twice, with each time having 2 consecutive convolutions. Then, it uses the green block three times, with each time having 3 consecutive convolutions. There is no max pooling after the last convolution.

form expression for the feature importance scores of $[\mathbf{GC}]$ as a function of the model's parameters. This is achieved by:

Proposition 2.1 (α coefficients for GradCAM, $V = 1$)**.** *Recall that the* $\mathbf{a}$ *vectors denote the non-rectified activation and* $\mathbf{W}$ *the weights of the linear part of* $[\mathbf{CNN}]$. *Then, for input* $\boldsymbol{\xi}$*, the* $[\mathbf{GC}]$ *coefficient* α *is given by*

$$\alpha = \frac{1}{hw}\sum_{i,j=1}^{h',w'}\sum_{i_1,\ldots,i_{L-1}=1}^{d_1,\ldots,d_{L-1}} \mathbb{1}_{\mathbf{a}^{(1)}_{i_1},\ldots,\mathbf{a}^{(L-1)}_{i_{L-1}}>0}\prod_{p=1}^{L}(\mathbf{W}^{(p)}_{i_p,i_{p-1}})^\top ,$$

where we set $i_0 := (i,j)$*,* $i_L = 1$*,* $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_\ell\times d_{\ell-1}}$ *are the weights of the* ℓ*-th hidden layer in* $\mathcal{F}$ *and* $\mathbf{a}^{(\ell)} \in \mathbb{R}^{d_\ell}$ *the pre-activation of the* ℓ*-th hidden layer.*

From Proposition 2.1, we immediately deduce a closed-form expression for GradCAM explanations. We note that Proposition 2.1 can be readily extended to an arbitrary number of filters $V > 1$, in which case the $\mathbf{a}$ and $\mathbf{W}$ should be interpreted as corresponding to the relevant $v \in [V]$. Using the closed-from coefficient of Proposition 2.1, we are able to describe precisely the behavior of GradCAM at initialization for our $[\mathbf{CNN}]$, specifically when the classifier part of our model comprises a single layer ($L = 1$). Our main result is:

Theorem 1 (Expected GradCAM scores, $L = 1$, masked $[\mathbf{CNN}]$)**.** *Let* $\boldsymbol{\xi} \in [0,1]^{H\times W}$ *be an input image. Let* $\mathbf{m} := \boldsymbol{\xi}_{i:i+k-1,j:j+k-1}$ *be the patch of* $\boldsymbol{\xi}$ *corresponding to index* $(i,j) \in [\![h]\!]\times[\![w]\!]$*. Assume that* h' *is even, and* $\mathbf{W}_{:,-\frac{h'}{2}:,:} = 0$*. Assume that the filter values and the non-zero weights are initialized i.i.d.* $\mathcal{N}(0,\tau^2)$*. Then, if the number of filters* V *is greater than* 20*, we have the following expected lower bound on the GradCAM explanation for pixel* (i,j)*:*

$$\begin{aligned}\mathbb{E}\left[[\mathbf{GC}]_{i,j}\right] &= \mathbb{E}\left[\sigma\left(\sum_{v=1}^{V}\alpha_v \mathbf{B}^{(v)}_{i,j}\right)\right]\\ &\geqslant \frac{V-20}{\sqrt{V}}\sqrt{\frac{h'w'}{16\pi}\frac{\tau^2}{hw}}\,\|\mathbf{m}\|_2\ ,\end{aligned}$$

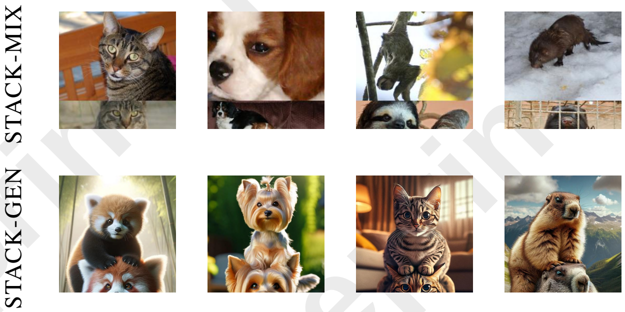

Figure 5: Sampled images from both of our datasets, *i.e.*, STACK-MIX and STACK-GEN.

where the expectation in the previous inequality is taken with respect to initialization of the filters and the remaining weights of the linear layer.

Setting $\mathbf{W}_{:,-\frac{h'}{2}:,:}$ to 0 disables the weights within $\mathbf{W}$ that are connected to the lower half part of the activation map $\mathbf{C}_{:,-\frac{h'}{2}:,:}$, effectively preventing $[\mathbf{CNN}]$ from accessing the lower half of $\mathbf{C}$. In turn, $[\mathbf{CNN}]$ does not see the lower half of ξ, up to side effects. The main consequence of Theorem 1 is that, when the number of filters associated to the class to explain is large enough, $[\mathbf{GC}]_{i,j}$ is positive in expectation if some pixels are activated in the receptive field associated to (i,j). Thus **GradCAM highlights all parts of the image where there is some "activity," even though this information is not used by the network in the end.** We illustrate Theorem 1 in Figure 3. The main limitation of this analysis is its focus on the behavior at initialization.

3 Experiments

We know ask the following question: are the consequences of Theorem 1 true after training, and for a more realistic model? To this extent, we train a CNN-based model which by construction cannot access some specified part of the input which we call the *dead zone* (see Figure 4). Clearly, since the dead zone does not influence the output, it should not contain positive model explanations. To test whether this is true, we create two datasets. Each item of the first one is composed of two images from ImageNet with the same label in both the seen and the unseen part of the image. The second dataset is built using generative models on the same categories with two objects in each image located in the seen and unseen part as well. We then check whether CAM-based methods wrongly highlight areas in the dead zone in Section.

Model definition. The CNN used in our experiments is a modification of a classical VGG16 architecture [Simonyan and Zisserman, 2015] which we call $[\mathbf{VGG}]$ (see Figure 4). The main modification is to forbid the network from seeing the dead zone in a very simple way: in the first dense layer $\mathbf{W}$, which has size 4096 × (256 × 14 × 14), we permanently set to 0 a band of height 9 corresponding to the lower weights. Formally, this means setting $\mathbf{W}_{:,:,-9:,:} = 0$, which is denoted in red above $\mathbf{W}$ in Figure 4. Effectively, we are building a wall that stops all information flowing from

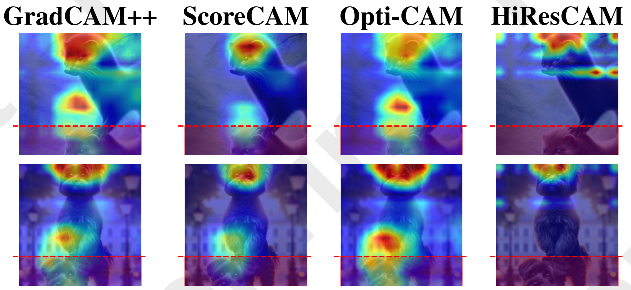

Figure 6: Saliency maps given by the considered CAM-based methods for [**VGG**]. With the notable exception of HiresCAM, they all highlight parts of images from STACK-GEN which are unseen by the network (this is denoted by the red, rectangular shape in the lower part of the image).

methods	STACK-MIX ↓	STACK-GEN ↓
GradCAM	22.7 ± 13.4	21.6 ± 11.6
GradCAM++	28.8 ± 8.1	28.5 ± 7.9
XGradCAM	23.8 ± 9.0	22.8 ± 9.0
ScoreCAM	19.9 ± 10.3	18.5 ± 10.6
Opti-CAM	32.7 ± 7.9	32.0 ± 7.8
AblationCAM	21.0 ± 9.9	20.8 ± 9.6
EigenCAM	51.7 ± 19.7	55.8 ± 21.6
HiResCAM	0.0 ± 0.0	0.0 ± 0.0

Table 1: Activity in the unseen part of the image, measured by $\mu \times 100$ for several CAM-based methods on both proposed datasets (only images in the validation set are considered).

the last convolutional layer to the remainder of the network. Since the weights $\mathbf{W}$ are directly connected to the final activation map $\mathbf{B} \in \mathbb{R}_+^{256\times14\times14}$, this masking effectively zeroes out the lower sections in each channel denoted by $\mathbf{B}_{:,-9:,:}$. We can trace back the zeroed activations in $\mathbf{B}$ to the preceding activation map $\mathbf{C}$, pinpointing the exact patches in $\mathbf{C}$ that correspond (after convolution) to the features observed in the zeroed activation of $\mathbf{B}$. Because of the side effects in the computation of convolutions, this area of $\mathbf{C}$ is slightly smaller: some pixel activation will still play a role in the model's prediction. Repeating this process until we reach the original image yields a dead zone of height 54 pixels, highlighted in red above $\boldsymbol{\xi}$ in Figure 4, which covers 24% of the image area.

We train [**VGG**] on Imagenet-1k [Deng *et al.*, 2009] using a classical training procedure. We compare the validation top-1 and top-5 accuracy of the VGG16 model found in the PyTorch repository. Our [**VGG**] without max pooling and no masking offers the same performance: 71.5% top-1 and 90.4% top-5 accuracy on the validation set. Indeed, our model [**VGG**] with masking has lower performance, which is expected as a fourth of the input image, $\boldsymbol{\xi}_{:,171:224,:}$, is unseen by the model. We obtain 66.5%, resp. 71.5%, top-1 and 87.0%, resp. 90.4%, top-5 accuracy on the validation set for our masked [**VGG**], resp. unmasked [**VGG**]. Nevertheless, we see that [**VGG**] is a **realistic network able to predict ImageNet classes with reasonable accuracy**.

New datasets. To assess how much CAM-based saliency maps emphasize irrelevant areas of an image, we introduce two new datasets in which we control the positions of the image elements using two techniques: cutmix [Yun *et al.*, 2019] and generative model. More precisely, we produce two datasets, called STACK-MIX and STACK-GEN. Where each image contains two objects, one in the bottom part of the image which is the dead zone for [**VGG**], and the second subject at the top of the image. Therefore, the subject at the center of the image will be mainly responsible for the top-1 predicted score by our masked [**VGG**].

Results. For our [**VGG**], we generate saliency maps from various CAM-based methods on our two datasets, STACK-MIX and STACK-GEN, using the predicted category for each example. We used publicly available implementations whenever possible. For each method, we measure how much of the CAM-based saliency maps emphasize the unseen part, *i.e.*, the dead zone. We use the metric μ defined for a upscaled saliency map $\mathbf{\Lambda} \in \mathbb{R}_+^{224\times224}$ as follows:

$$\mu(\mathbf{\Lambda}) := \frac{\|\mathbf{\Lambda}_{171:224,:}\|_2}{\|\mathbf{\Lambda}\|_2}\,, \tag{1}$$

where $\|\cdot\|_2$ is the ℓ^2-norm and the lower part of the image $\boldsymbol{\xi}_{:,171:224,:}$ is unseen by our [**VGG**]. We note that for a saliency map $\mathbf{\Lambda}$, the lower $\mu(\mathbf{\Lambda})$, the better. The results can be found in Table 1, and Figure 6. We observe that every CAM-based methods, except HiResCAM, highlights unseen parts of an image to some extent. Moreover, the observation are consistent over both datasets.

4 Conclusion

In this paper, we looked into several CAM-based methods, with a particular focus on GradCAM. We showed that they can highlight parts of the input image that are provably not used by the network. This was also showed theoretically, looking at the behavior of GradCAM for a simple, masked CNN at initialization: the saliency map is positive in expectation, even in areas which are unseen by the network. Experimentally, this phenomenon appears to remain true, even on a realistic network trained to a good accuracy on ImageNet.

Acknowledgments

This work was funded in part by the French Agence Nationale de la Recherche (grant number ANR-19-CE23-0009-01 and ANR-21-CE23-0005-01). Most of this work was realized while DG was employed at Université Côte d'Azur and RS at Centrale Méditerranée. We thank Jenny Benois-Pineau for her valuable insights.

References

[Adebayo *et al.*, 2018] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, 2018.

[Benítez *et al.*, 1997] José Manuel Benítez, Juan Luis Castro, and Ignacio Requena. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, 1997.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[Draelos and Carin, 2021] Rachel Lea Draelos and Lawrence Carin. Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks. *arxiv preprint 2011.08891*, 2021.

[Fukushima, 1980] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980.

[Garreau and Mardaoui, 2021] Damien Garreau and Dina Mardaoui. What does LIME really see in images? In *International Conference on Machine Learning*. PMLR, 2021.

[Ghorbani *et al.*, 2019] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of Neural Networks is Fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[Heo *et al.*, 2019] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling Neural Network Interpretations via Adversarial Model Manipulation. In *Advances in Neural Information Processing Systems*, 2019.

[Kindermans *et al.*, 2019] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (Un)reliability of saliency methods. *Explainable AI: Interpreting, explaining and visualizing deep learning*, 2019.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 1998.

[Linardatos *et al.*, 2021] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 2021.

[Lipton, 2018] Zachary C. Lipton. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. 2018.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, 2016.

[Selvaraju *et al.*, 2017] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, 2013.

[Taimeskhanov *et al.*, 2024] Magamed Taimeskhanov, Ronan Sicre, and Damien Garreau. CAM-Based Methods Can See Through Walls. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024.

[Yun *et al.*, 2019] S. Yun, D. Han, S. Chun, S. Oh, Y. Yoo, and J. Choe. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[Zhang *et al.*, 2021] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

[Zhang *et al.*, 2023] Hanwei Zhang, Felipe Torres, Ronan Sicre, Yannis Avrithis, and Stephane Ayache. Opti-CAM: Optimizing saliency maps for interpretability. *arXiv preprint arXiv:2301.07002*, 2023.

[Zhou *et al.*, 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.