Don’t Augment Me: On the Robustness of Saliency Methods

Kemal Erdem*† **Piotr Mazurek**†

†Department of Computer Science and Management
Wrocław University of Science and Technology
kemal@erdem.pl

Abstract

The recent advent of explainable AI methods may have given deep learning researchers a false sense of understanding of the models. Attribution methods that are not reliable enough to coherently explain models with varied inputs are used in many scientific articles without fully understanding how they work and what their limitations are. This article shows that outcomes of currently existing saliency methods can vary significantly for the same input when we apply seemingly non-invasive augmentations to it. For an input RGB image, we apply augmentations that aim to simulate the transformations that can happen to the image in real-world scenarios, such as sharpening, normalization, color boosting, and rotation. The used augmentations have a rather insignificant impact on the model predictions, but at the same time, have a significant impact on the attributions returned by a number of saliency methods. We argue that such methods, while being useful for finding apparent confirmation for the reliability of a classification model, may be of little use as far as real-world applications are concerned.

1 Introduction

As deep learning methods mature, more and more industries try to use these algorithms. Among the applications that are especially interesting are those in safety-critical domains such as cybersecurity or healthcare. One of the major problems with applying deep learning is the lack of transparency caused by the complicated nature of deep learning models and the millions of parameters present in neural networks. Explainable AI methods give an apparent answer to these problems. The authors of those methods often promise that their techniques will provide a clear answer to the question of how a model’s decision was made, by applying some variation of backprop-based operations.

This work shows that popular XAI methods such as Integrated Gradients [1], Saliency [2], Guided GradCAM [3, 4], Gradient SHAP [5], Guided Backpropagation [6] should not be trusted because of the lack of consistency in the attributions produced by those methods. We argue that if attributions for the images differ significantly for relatively simple natural images of food or dogs, it should not be trusted to explain models working in safety-critical domains. We study the effects of image augmentations on the saliency maps returned by XAI methods. We measure the similarities between attributions returned for the same image before and after applying an augmentation.

The original integrated gradients paper [1] is already referenced nearly 1500 times by various researchers as of May 2021. It has been applied for explaining models used in healthcare [7, 8, 9, 10, 11, 12, 13], genes analysis [14], hate speech detection [15], drug discovery [16], and other

[*] Personal website https://erdem.pl

Preprint. Under review.

areas [15, 17]. Similarly, other saliency methods find applications in various fields like GradCAM in Covid-19 X-Ray analysis [18, 19] and fraud detection [20], or Saliency in Parkinson's Disease recognition [21].

Despite the controversy related to using the saliency methods, various authors seem to use those without any more profound reflection. The practice is especially concerning in the case of healthcare and cybersecurity systems. This trend seems to be only accelerating as accessibility of easy-to-use XAI libraries increases.

2 Related work

Hooker et al.[22] have shown that several of widely used saliency methods returns attributions not better than a simple edge detector or even random noise. They achieved it by removing a subset of pixels supposedly most relevant to the final prediction (most important according to popular attribution methods) and retraining models on an updated dataset without the "relevant" pixels. The assumption is that - assuming the XAI method correctly identifies the relevant pixels - the model performance should drastically drop once the critical part of the information has been removed. Yet this does not happen. Surprisingly, removing pixels suggested by popular attribution methods (among others Integrated Gradient[1] and Guided Backpropagation[6]) had less impact than removing random pixels or pixels found by a simple edge detector (Sobel filter). The model performed better when random pixels were removed from the dataset, than when the pixels were chosen using saliency methods. This suggests that the tested saliency did not indicate the relevant pixels.

Another case against attribution methods is made by Ghorbani et al. [23], who discuss attacks using adversarial perturbations to produce different interpretations of the same image. Their paper shows that we can generate such adversarial perturbations that produce perceptively indistinguishable inputs with the same prediction, but the attribution differs by a considerable margin from the original attribution. Our work also focuses on the effect of input perturbation, but instead of searching for adversarial perturbation, we apply augmentations that aim to simulate real-world image processing that happens in modern cameras (see Fig. 1) and check how they influence attribution.

There were some efforts on trying to measure the quality of attribution methods [24, 25]. The first paper[24] uses so-called *region perturbation* to progressively remove information from the image at the specified location. Their measurement checks the relative change of the explanation with each iteration of the algorithm. The idea is that when perturbing regions, chosen by a good attribution method, it should have a higher impact. The second paper[25] introduces *Sensitivity* and *Infidelity* measures. Both of those measures are applying a perturbation to the input image. Sensitivity tries to check how much the attribution change when "insignificant" noise is used. Infidelity uses "significant" noise and checks how it affects models' score based on the attribution values. Measuring XAI methods is still an active area of research, and the current results are still far from perfect.

3 Attribution methods

Our work has focused on the most popular attribution methods available in modern XAI libraries (e.g., Captum [26]). Selecting methods based on availability and popularity was dictated by an assumption that those methods will be the first choice when trying to explain models. This is a brief description of the methods used in the experiments and the idea of how an attribution method works.

An *attribution method* is defined as $A : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ where $m \times n$ is the dimension of the input matrix $x \in \mathbb{R}^{m \times n}$. Provided explanation corresponds to the "importance" of element from the input matrix for a given *class* C and *model* $F : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{C}$.

Saliency (also known as "Vanilla Gradient") calculates gradient values $A_{Saliency}(x) = \frac{\partial F(x)}{\partial x}$ [2, 27] that shows how much each element of the input, contributes to the predicted class.

Guided Backpropagation (GBP) [6] uses additional 'deconvolution' approach known from Deconvolutional Networks[28] to compute attribution values. GBP does not compute a true gradient but rather an imputed version of it.

Integrated gradients (IG) [1] is defined as $A_{\mathrm{IG}}(x) = (x - \bar{x}) \times \int_0^1 \frac{\partial F(\bar{x} + \alpha(x - \bar{x}))}{\partial x} d\alpha$ where $\bar{x}$ is a baseline representing absence of a feature in the input x.

Guided GradCAM [3, 4] is an extension of the standard GradCAM method and calculates attribution using Hadamard product of GPB and GradCAM. $A_{guided-gradcam} = A_{gradcam} \odot A_{gpb}$.

Gradient SHAP is based on the original *SHAP* [5] method and works by approximating SHAP values based on expected gradients [29]. It uses the randomly generated baselines and selects random point from the path between that baseline and the input modified with a Gaussian noise. Then it computes the gradient with respect to those random points and multiply the result by the difference between the input and the baseline $\nabla F \times (x - \bar{x})$. Returned attributions are an approximation of the SHAP values.

Noise Tunnel [30, 22, 31] is applied on top of the attribution methods. It uses Smoothgrad [29] to combine multiple attributions from the method. Each attribution is calculated from the same input with added Gaussian noise. For a given method A, it computes $A_{NoiseTunnel}(x) = \frac{1}{N} \sum_{i=1}^{N} A(x + g_i)$, where noise $g_i \sim \mathcal{N}(0, \sigma^2)$ is drawn independently and identically distributed from a Gaussian distribution.

4 Experiments

In order to measure the volatility of the attributions returned by various saliency methods, we measure how the attributions change after augmentation is applied to the images. For each saliency method, we compare the baseline attributions with attributions of a transformed image. The differences in attributions are measured using Structural Similarity Index Measure (SSIM) [32]. All experiments took around 300h of computation using GeForce GTX 1080 Ti GPU.

4.1 Datasets

We decided on training the classification models on 5 diverse, publicly available datasets.

Edible plants [33] is an image classification dataset of different types of wild edible plants. It contains 6875 images of 62 categories. For experiments, we follow the official train/test split.

Food 101 [34] is an image classification dataset of different types of food. It contains 101,000 images of 101 categories. For experiments, we follow the official train/test split.

Marvel Heroes [35]is an image classification dataset of characters from the Marvel universe. It contains 3,035 images of 8 categories. For experiments, we follow the official train/test split.

Plants [36] is an image classification dataset of different species of plants. It contains 19,900 images of 99 categories. For experiments, we follow the official train/test split.

Stanford Dogs [37] is an image classification dataset of dogs of different breeds. It contains 20,580 images of 120 categories. For experiments, we use an 80:20 train/test split with stratification.

4.2 Classification models

For each dataset, we trained 3 models: ResNet 18 [38], DenseNet 121 [39] and EfficientNet B0 [40]. All models are pretrained on the ImageNet[41] and fine-tuned on the given dataset. Fine-tuning is done using Stochastic Gradient Descent (SGD) with momentum [42] and Cross-Entropy loss function. Each model was trained for 15 epochs with a decrease of the learning rate (from *0.001* to *0.0001*) after the 7th epoch. To test a broad range of models, we decided to create five versions of each training set (*{20%, 40%, 60%, 80%, 100%}* of the entire training set). This way, we are able to lower the models' performance for the specific task. The total number of trained models was 75.

4.3 Augmentations

We applied 4 rotations and 5 more general transformations for each tested image, nine transformations in total. Methods aim to simulate the changing conditions that can happen to the image during capture using a smartphone camera or during the image post-processing in a popular image editing software. The minor rotations or the slight changes of colors should have minimal impact on the attribution returned by a well-designed XAI method. We chose filters that closely resemble ones available in popular camera apps. The effects of the filters can be seen in Fig. 4.

Rotations are applied to every sample in the test set. The reason to use them was to simulate the change that can easily happen to the image during capture. For each image, we applied 4 rotations $\{-30, -15, 15, 30\}$ degrees. While comparing attributions of rotated images with the original attribution, the latter was rotated in the same way.

Freaky Details [43] is a method that aims to enhance the details of an image (local contrast) by setting the surface blurred layer to a vivid light mode over the base image. Parameters used for this augmentation: *Amplitude: 2, Scale: 10, Iterations: 1.*

Local Normalization [44] is a method that smoothes the pixels in the local neighborhood. Neighboring pixels are shifted based on the local maximum and minimum. Parameters used for this augmentation: *Amplitude: 8, Radius: 10.*

Boost chromaticity [45] is a method that is slightly changing the color of pixels by increasing the colorfulness parameter of chromaticity. Parameters used for this augmentation: *Amplitude: 90%, Color Space: YCbCr.*

Mighty details [46] is another method of local contrast enhancement. It boosts colors and "paintifies" an input image. Parameters used for this augmentation: *Amplitude: 25, Details Amount: 1, Details Scale: 25, Details Smoothness: 1, Channels: YCbCr*

Sharpen [47] is a method that sharpens the image by inverse diffusion and shock filters methods. Parameters used for this augmentation: *Amplitude: 300.*

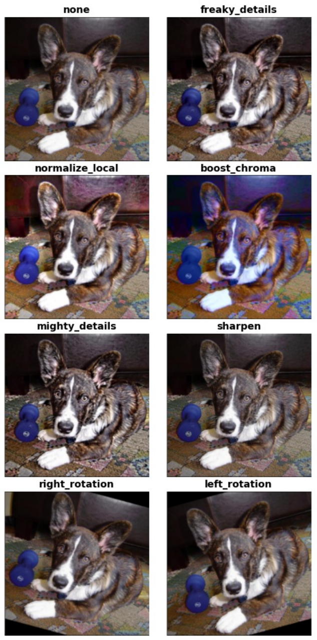


Figure 1: Examples of real-world augmentations applied on the input image

4.4 Measuring attribution structural similarities

To measure the variability of each saliency method, we rely on the structural similarity (SSIM) [32] index. We compute SSIM between attributions for the base image and the one produced for the transformed image. If the transformation applied on the input image is a rotation, we apply the same rotation on the attribution from the base image. This additional transformation of the base image attribution allows us to ignore most of the dissimilarities caused by the rotation and calculate the value of SSIM for attributions with the same rotation.

The final average SSIM score (Fig. 2) is calculated using only those examples where the prediction for the original image is different from the prediction for the augmented image by less than $threshold = 0.05$. This filtering process ensures that only augmentations with insignificant impact on model performance are used for calculating the similarity of two attributions. We assume that if the prediction is not affected, neither should the attribution be. Therefore, if attributions differ, then the tested XAI is probably not working as expected.

SSIM values closer to 1.0 should indicate that the XAI method performs well for the augmented image; therefore, the method is better suited for real-world usage. Quantifying a human visual perception is a complex problem. We are aware that using SSIM is not going to produce ideal results (e.g., for Fig. 3 despite SSIM being close to 1.0, the two attributions differs considerably), but as a quantitative measurement, it allows us to compare results from the entire dataset reliably.

We use SSIM with a window of size 11 and calculate the value range as a maximum attribution value for a given experiment. Changing the value range allows us to normalize the results for the methods that usually provide low absolute attribution values (e.g., Guided GradCAM). The visualization of the Guided GradCAM attributions and the related SSIM score is shown in Figure 3.

4.5 Results

Studied methods achieved various mean similarity scores, ranges from 0.68 for Gradient SHAP to 0.95 for Guided GradCAM (see Fig. 2). When comparing only the mean values, we can think that Guided GradCAM achieves almost perfect results (closer to 1.0). This particular method's high mean SSIM value might be due to less noise produced when generating attribution. Because Guided

GradCAM creates attributions that resemble edge detectors[22], even if that attribution changes between the base and augmented image, most pixels remain the same (close to 0).

The mean value itself is not sufficient to describe methods, and we have to look at the standard deviation (std) of the values. High std values inform us that there is a lot of varied examples. In some of them, the augmentation has a less significant impact on the attribution, but there are many cases when the difference in similarity is substantial. Once again, Guided GradCAM beats other methods. Because of that, for visualization purposes, we are going to use this method. We will assume that if the best method can change its attribution, then methods that scored worse are not better.

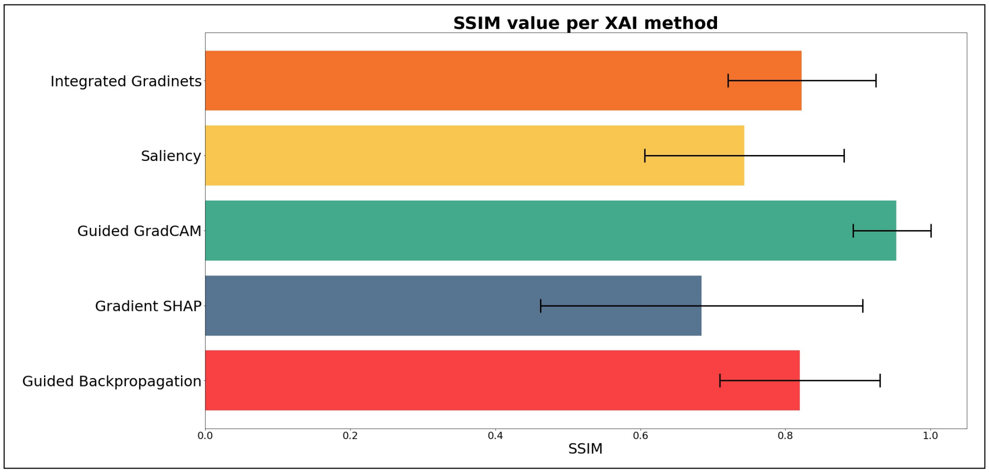
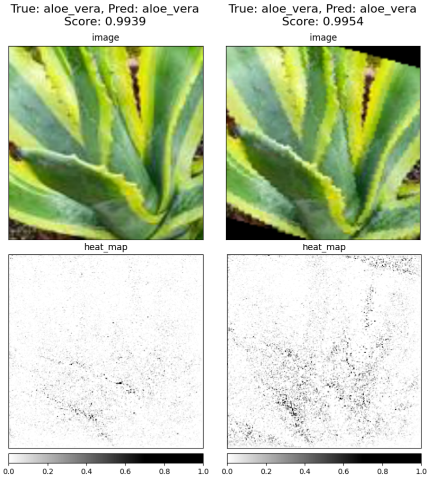


Figure 2: Average SSIM values per attribution method. Each bar represents a methods' mean value of SSIM. This mean values excludes examples when classification score of the augmented image is outside the $threshold$ of non-augmented image. Because score is independent from the attribution method, all mean values are calculated from exactly the same images and augmentations.

Visualizations are generated using *Guided GradCAM* attributions of the input image. This method is selected because it achieved the best mean SSIM from the five that we have tested. All the examples picked for the visualization are selected to be within the $threshold$ of the class score from a base image. This is more restrictive than calculating mean SSIM values but allows a qualitative comparison of the same augmentation types without filtering out those examples that modify the final model score.

To better understand how the SSIM measure works, we can compare scores with attributions on which those scores were achieved. Usually, the higher scores indicated more similar attributions, and we can see that when comparing attributions from Figure 4. If we look at the cardigan example and compare the base image (none) with *normalize_local* augmentation, we can see that attribution is pointing to the different parts of the dog. SSIM value for that pair is *0.8762*. Using the same base image, we can compare its attribution with attribution generated when applying *freaky_details* augmentation. SSIM value for that pair is *0.9163* and this attribution is more similar to the original one.

As shown in Figure 4, applying common filters to the input image changes the input attributions signif-



(a) Original image (b) Rotation of -15°

Figure 3: Visualisation of attributions for example image from Plants [36] tested on EfficientNet B0 [40] trained on 100% of the training dataset. SSIM value for this pair: **0.9617**.

icantly. Changes to the attributions are nondeterministic; therefore, original attributions cannot be relied on in real-world scenarios. Nondeterministic behavior can be seen in the third column (Fig. 4). In the case of the *harebell*, augmentation causes the attribution to contain more "noise" (compared to the original one). When we look at the *pancakes*, then attribution is more focused on the edges than attribution from the original input. The *cardigan* case (Fig. 4, *cardigan's* original attribution vs. attribution for local normalization) shows us that the same augmentation shifts the attributions towards the top of the image. This might be especially dangerous when trying to understand what the model emphasizes when predicting *cardigan* class.

Some of the applied filters cause the attribution to be shifted into looking more like a noise (Fig. 4, *harebell's* attribution for freaky details filter) and lose the visual information that is important for a human to understand the models' prediction.

Augmentation might also cause the method to show less attribution on the augmented image (Fig. 4, *pancake's* attribution for sharpening filter) by lowering its absolute values.

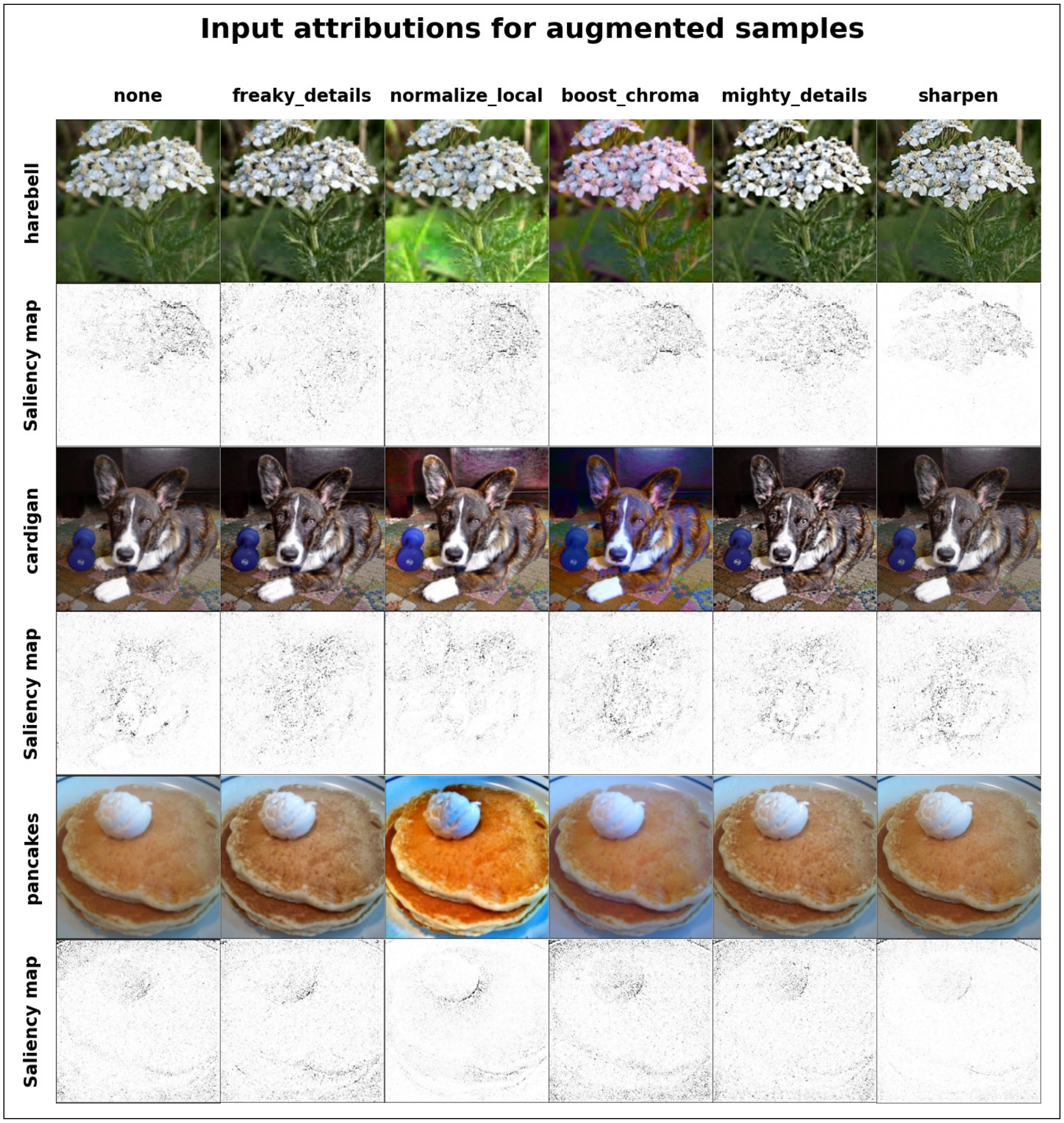


Figure 4: **Attribution comparison of inputs with applied filters.** Figure shows three distinct images (*harebell_flower*, *cardigan*, *pancakes*) with different filters. All augmented images in each row are classified using the same model and achieve scores within the $threshold$ from the score of the image with the *"none"* label.

Similar to filters (Fig. 4), rotations (Fig. 5) seem to modify the results produced by the tested XAI methods. For the picture of the *harebell* (Fig. 5), after each rotation, the attributions change significantly. Instead of focusing on flower petals, the tested XAI method seems to produce attributions resembling random noise rather than a coherent explanation of model behavior.

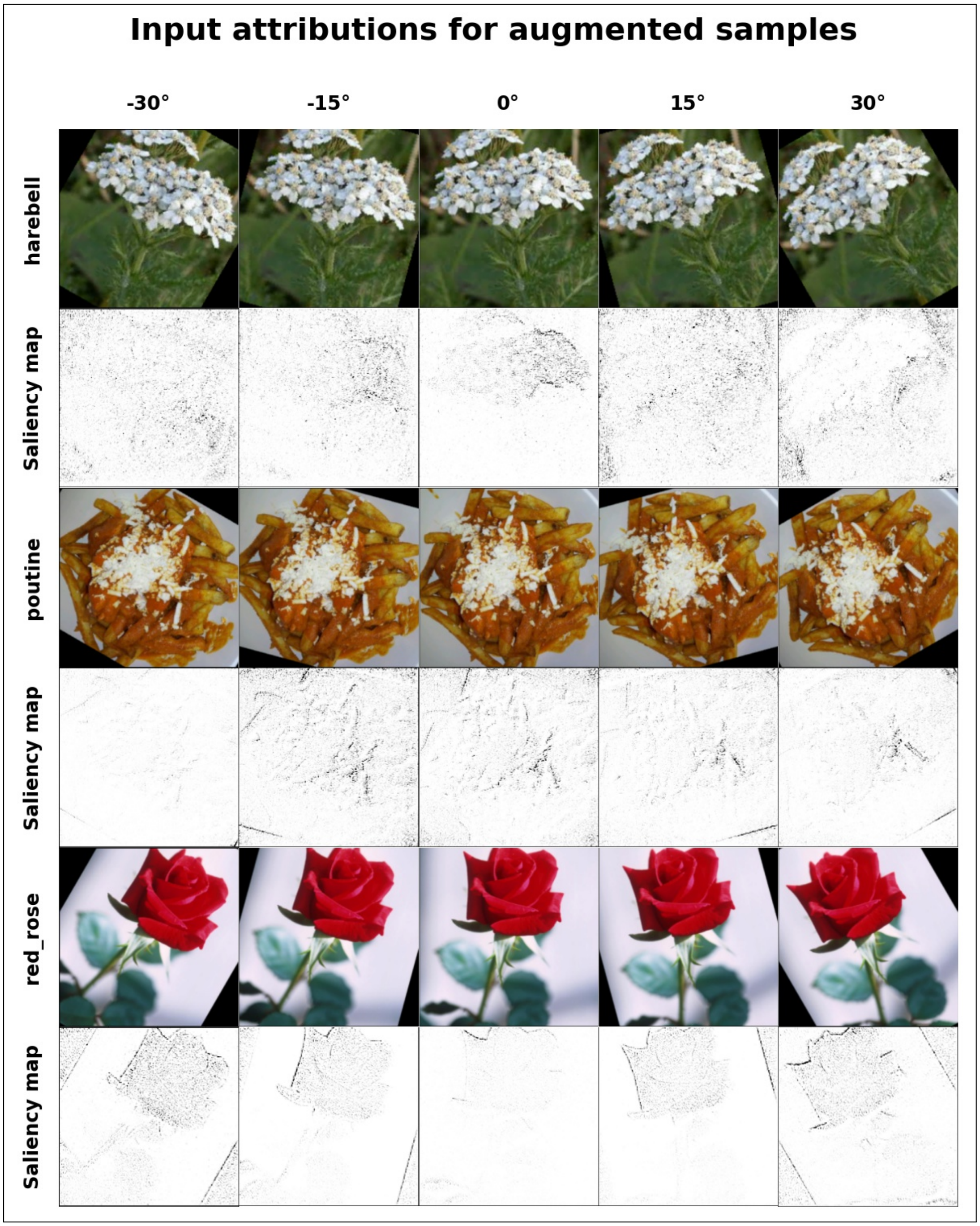


Figure 5: **Attribution comparison of rotated inputs.** Figure shows three distinct images (*harebell_flower*, *poutine*, *red_rose*) with different rotations. All augmented images in each row are classified using the same model and achieve scores within the $threshold$ from the score of the image with the *"0°"* label.

The concerning saliency method outcome can be seen in the *poutine* image example (Fig. 5). While for some rotations $(-15, 15, 30)$ the XAI method does not seem to change the produced attributions significantly, for a -30 rotation it is completely lost, unable to reveal the features based on which the model prediction was made. Surprisingly the opposite seems to be the case for a *red-rose* image (Fig. 5). For the original image, according to the XAI method, only the tips of the petals are important for the prediction, while after any rotation of the XAI method recognizes that the whole rose petals are equally crucial for the final prediction. All of those augmented images have almost identical score as a base image when predicting the class.

An example of lack of consistency in the returned attributions is particularly evident in Fig. 3. With a slight (15 degrees) rotation of the picture, the attribution changes entirely even though the prediction hardly changes at all (*aloe vera*: 0.9939 for original and *aloe vera*: 0.9954 for rotated). The reliable XAI method should not have such significant changes in generated attributions after a slight image rotation.

5 Discussion

As deep learning models become better and better every year, more industries apply those methods. The increased use of deep learning has resulted in growing concern over the problem of explaining such models, especially in safety-critical systems and in industries with a history of unfair biases.

In recent years the problem of the explainability is raised particularly often. Deep learning packages developers and public cloud providers are doing their best to provide the right tools for developers to better understand their models' behavior. However, as shown in the number of papers and this work, the existing methods are far from being perfect when the reliability of the models' explanation is considered. The attributions may change drastically after a minor change to the input image is applied. Some even argue that simple edge detectors often do a better job than many of the existing attribution methods for images [30, 22].

Common practice in the Deep Learning community is to validate a model by checking which part of the image has a higher attribution for a given class. As shown in Figure 4 and the *cardigan* example (attributions for an original image vs. attributions for local normalization), checking only one example under one condition, might provide a developer of the model with a piece of insufficient information on what part of the image is relevant to predict selected class.

The easy access to the explainable AI methods, in our opinion, may cause harm if used without caution. As more and more non-experts use those methods, they gain a false sense of security and false trust in the system, which may fail when they least expect it. Lowering the technical level required to use these methods will only exacerbate this problem in the coming years when better and better tools will be implemented, and more and more non-specialists will use them.

It does not help when a top-tier public cloud provider encourages the usage of Integrated Gradients[1] to explain one's image model [48] in the official guideline for non-experts. In our opinion, providers of XAI technologies should put much more effort into presenting the users of those technologies with the limitations of XAI tools and informing them of potential pitfalls.

6 Summary

The purpose of the experiments was to influence method researchers and encourage them to put more emphasis on showing the potential imperfections in their methods' behavior. We are presenting a potential direction of additional sanity checks for such methods. This approach could be applied almost effortlessly to already existing implementations.

Our results show that the attributions returned by several XAI methods can be affected by seemingly irrelevant transformations of images. Simple image augmentations such as rotation sharpening, normalization, or color boosting while having close to none impact the model accuracy, often significantly impact the attributions. If an insignificant change to the original image has such a massive impact on the attribution, then the method should probably not be used to explain the behavior of DL models in the safety-critical domains.

Broader Impact

In the article, we show the variability and the inconsistency of the currently existing saliency methods. We show that even if used XAI method seems to be working for some cases, it may not work for others. Therefore we hope that our work will encourage authors to include both types of XAI outcomes in their works and promote a broader discussion regarding AI safety and the fact that the currently widely-used saliency methods may not be enough to explain deep learning models coherently.

References

[1] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. 2017.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. 2014.

[3] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? 2017.

[4] 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. 2019.

[5] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 2017.

[6] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. 2015.

[7] Nils Rethmeier, Necip Oğuz Şerbetci, Sebastian Möller, and Roland Roller. Efficare: Better prognostic models via resource-efficient health embeddings. *medRxiv*, 2020.

[8] John-William Sidhom, Ingharan J Siddarthan, Bo-Shiun Lai, Adam Luo, Bryan C Hambley, Jennifer Bynum, Amy S Duffield, Michael B Streiff, Alison R Moliterno, Philip Imus, et al. Deep learning for diagnosis of acute promyelocytic leukemia via recognition of genomically imprinted morphologic features. *NPJ precision oncology*, 5(1):1–8, 2021.

[9] Andreas Kleppe, Ole-Johan Skrede, Sepp De Raedt, Knut Liestøl, David J Kerr, and Håvard E Danielsen. Designing deep learning studies in cancer diagnostics. *Nature Reviews Cancer*, pages 1–13, 2021.

[10] Wei Xiao, Xi Huang, Jing Hui Wang, Duo Ru Lin, Yi Zhu, Chuan Chen, Ya Han Yang, Jun Xiao, Lan Qin Zhao, Ji-Peng Olivia Li, et al. Screening and identifying hepatobiliary diseases through deep learning using ocular images: a prospective, multicentre study. *The Lancet Digital Health*, 3(2):e88–e97, 2021.

[11] Toon Van Craenendonck, Bart Elen, Nele Gerrits, and Patrick De Boever. Systematic comparison of heatmapping techniques in deep learning in the context of diabetic retinopathy lesion detection. *Translational Vision Science & Technology*, 9(2):64–64, 2020.

[12] Behrooz Mamandipoor, Fernando Frutos-Vivar, Oscar Peñuelas, Richard Rezar, Konstantinos Raymondos, Alfonso Muriel, Bin Du, Arnaud W Thille, Fernando Ríos, Marco González, et al. Machine learning predicts mortality based on analysis of ventilation parameters of critically ill patients: multi-centre validation. *BMC Medical Informatics and Decision Making*, 21(1):1–12, 2021.

[13] Anna Lind, Ehsan Akbarian, Simon Olsson, Hans Nåsell, Olof Sköldenberg, Ali Sharif Razavian, and Max Gordon. Artificial intelligence for the classification of fractures around the knee in adults according to the 2018 ao/ota classification system. *PloS one*, 16(4):e0248809, 2021.

[14] Ben Liu, Meng Zhou, Xiangchun Li, Xining Zhang, Qinghua Wang, Luyang Liu, Meng Yang, Da Yang, Yan Guo, Qiang Zhang, et al. Interrogation of gender disparity uncovers androgen receptor as the transcriptional activator for oncogenic mir-125b in gastric cancer. *Cell death & disease*, 12(5):1–22, 2021.

[15] György Kovács, Pedro Alonso, and Rajkumar Saini. Challenges of hate speech detection in social media. *SN Computer Science*, 2(2):1–15, 2021.

[16] José Jiménez-Luna, Francesca Grisoni, Nils Weskamp, and Gisbert Schneider. Artificial intelligence in drug discovery: Recent advances and future perspectives. *Expert Opinion on Drug Discovery*, pages 1–11, 2021.

[17] Maor Asif and Yaron Orenstein. Deepselex: inferring dna-binding preferences from ht-selex data using multi-class cnns. *Bioinformatics*, 36(Supplement_2):i634–i642, 2020.

[18] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, 140:110190, 2020.

[19] Kaoutar Ben Ahmed, Gregory M Goldgof, Rahul Paul, Dmitry B Goldgof, and Lawrence O Hall. Discovery of a generalization gap of convolutional neural networks on covid-19 x-rays classification. *IEEE Access*, 2021.

[20] Duygu Sinanc, Umut Demirezen, and Şeref Sağıroğlu. Explainable credit card fraud detection with image conversion. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 10(1):63–76, 2021.

[21] Theerasarn Pianpanit, Sermkiat Lolak, Phattarapong Sawangjai, Thapanun Sudhawiyangkul, and Theerawit Wilaiprasitporn. Parkinson's disease recognition using spect image and interpretable ai: A tutorial. *IEEE Sensors Journal*, 2021.

[22] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. 2019.

[23] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[24] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[25] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I Inouye, and Pradeep Ravikumar. On the (in) fidelity and sensitivity for explanations. *arXiv preprint arXiv:1901.09392*, 2019.

[26] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

[27] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. 2010.

[28] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 2013.

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. 2017.

[30] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. 2020.

[31] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. 2018.

[32] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. 2004.

[33] Gabriel Verzea. Edible wild plants. https://www.kaggle.com/gverzea/edible-wild-plants, 2018. Accessed: 2021-05-01.

[34] ETH. Food images (food-101). https://www.kaggle.com/kmader/food41, 2018. Accessed: 2021-05-01.

[35] Marvel heroes. https://www.kaggle.com/hchen13/marvel-heroes, 2019. Accessed: 2021-05-01.

[36] Muhammad Jawad, Muhammad Safwan, Hamza Usman, and Rimsha Khan. Plants dataset[99 classes]. https://www.kaggle.com/muhammadjawad1998/plants-dataset99-classes?select=Plant_Data, 2020. Accessed: 2021-05-01.

[37] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Stanford dogs dataset. https://www.kaggle.com/jessicali9530/stanford-dogs-dataset, 2019. Accessed: 2021-05-01.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. corr abs/1512.03385 (2015), 2015.

[39] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[41] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[43] Freaky details filter. https://natron.readthedocs.io/en/rb-2.3/plugins/eu.gmic.FreakyDetails.html. Accessed: 2021-05-01.

[44] Local normalization filter. https://natron.readthedocs.io/en/rb-2.3/plugins/eu.gmic.LocalNormalization.html. Accessed: 2021-05-01.

[45] Boost chromacity filter. https://natron.readthedocs.io/en/rb-2.3/plugins/eu.gmic.BoostChromaticity.html#gmic-boost-chromaticity-node. Accessed: 2021-05-01.

[46] Mighty details filter. https://natron.readthedocs.io/en/rb-2.3/plugins/eu.gmic.MightyDetails.html. Accessed: 2021-05-01.

[47] Sharpening filter. https://natron.readthedocs.io/en/rb-2.3/plugins/eu.gmic.SharpenOctaveSharpening.html. Accessed: 2021-05-01.

[48] Google Cloud Blog. Introduction to vertex explainable ai for vertex ai. https://cloud.google.com/vertex-ai/docs/explainable-ai/overview#ig. Accessed: 2021-05-01.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.

- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the limitations of our metric SSIM in 4.4
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work probably don't have negative work since we only show the downsides of currently existing methods
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See the similarities assumptions in 4.4
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] `https://github.com/burnpiro/xai-correlation`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See 4.1 and Github Readme and Wiki
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] In our case, the exact model performance is not the point of experiments. Therefore we do not provide the error bars.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.1 Datasets
   (b) Did you mention the license of the assets? [No] We use only datasets updated publicly on kaggle
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We rely on already existing datasets
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We assumed that publicly datasets are published with a consent
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We assume that the authors of the publicly available datasets took care of that
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]