UNIVERSITY OF GRONINGEN

RESEARCH INTERNSHIP

Analysis of Feature Attribution Methods in Explainable AI

Author:
Ayça Avcı (s4505972)

Principal project advisor:
Prof. Dr. Ir. G. GAYDADJIEV

Second project advisor:
R. BRANDT, M.Sc.

June 27, 2021

Abstract

In the last decade, neural network-based classifiers have started to perform impressively well. Their computation often involves an intransparent process, and explaining the output of the networks remains a challenge. This sparked the novel field *eXplainable Artificial Intelligence (XAI)* which aims to disclose this intransparency. Several XAI methods called *feature attribution methods* have been proposed, which show for a prediction how much was contributed to a prediction by each input feature. The purpose of this paper is to compare five state-of-the-art feature attribution methods, namely Vanilla Gradients [30], SmoothGrad [31], XRAI [18], Grad-CAM [28], and BlurIG [37] using different evaluation metrics such as MoRF [23] [27], smoothness, similarity to edge map [6], invariance to model weights [6], and invariance to data labeling [6]. Results showed that the metrics MoRF and invariance to data labeling advocate that BlurIG produces a better explanation. However, similarity to edge map and invariance to model weights metrics argue that Vanilla Gradient provides a better explanation. On the other hand, smoothness metric points out that Grad-CAM and Grad-CAM$_S$ have better explanations compared to other methods. Therefore, we conclude that it is currently not possible to explicitly say which method is best since the metric results do not agree on which method provides the best explanations.

Index Terms—explainable artificial intelligence, feature attribution, saliency maps, vanilla gradients, smoothGrad, XRAI, grad-CAM, blurIG.

Acknowledgments

I would like to thank expert reviewer Remi Brandt, M.Sc. for his insightful comments, suggestions, and contributions in Section 4 and Section 6, and the guidance that made this research possible.

Contents

1 Introduction

Explainable AI (XAI) is artificial intelligence (AI) where the produced results are understandable by humans. In contrast, a "black box" in machine learning is the concept where the decisions made by the AI cannot be explained by its designers [14]. The relevance of XAI is important, even though it is not necessitated by any legal or regulatory requirements. For example, XAI can help end-users understand and trust that good decisions are being made by the AI, thus improving the user experience of some product or service. Therefore, XAI aims to understand the past, present, and future actions of the AI, and understand how and why they were made [16]. These characteristics affirm and challenge the existing knowledge and allow new assumptions to be made [25].

Feature attributions show how much was contributed by each feature in the model to the predictions for each given instance. Tabular data is used for feature attributions, as well as built-in visualization capabilities concerning image data. The change in prediction value due to the feature is shown by the attribution data, relative to the baseline value that is specified.

1.1 Advantages and use cases

Looking at the specific examples and aggregating feature attributions across the training set, deeper insight can be gotten about how the model works. In below, some of the advantages and use cases are presented:

- **Debugging models:** Feature attribution methods may identify issues in the data that standard model evaluation methods would usually overlook. For instance, an image pathology model has achieved doubtfully good results on a test set of chest X-Ray images [5]. Feature attribution methods disclosed that the model's high accuracy depends on the radiologist's pen marks in the image [4].
- **Optimizing models:** Identifying and removing less significant features, which might result in more efficient models.

1.2 Conceptual limitations

- For an individual prediction, investigating an attribution may offer good insight, but the insight might not be generalized to the whole class for that specific instance, or the whole model, since attributions are only specific to individual predictions. Attributions should be aggregated over subsets over the data set, or whole data set in order to get more generalizable insight [4].

- Sometimes, feature attributions do not clearly indicate whether issues arise from data that model is trained, or from the model itself even if they can help with debugging the model [4].
- As in complex model predictions, feature attributions methods are also vulnerable to adversarial attacks [4].

1.3 Contributions

To summarize, this research focuses on the followings:

1. Giving an overview of feature attribution methods that are used for image classification.
2. Comparison of five current state-of-the-art feature attribution methods and evaluation of them using a proper evaluation framework, which includes the following metrics: MoRF, smoothness, invariance to model weights, invariance to data labeling, and similarity to edge map.

1.4 Outline

In Section 1, a brief overview of what is explainable AI and feature attribution, use cases and advantages of feature attribution methods and conceptual limitations of them, focus of this research internship are provided. In Section 2, an overview of previously developed feature attribution methods and evaluation metrics are presented. In Section 3, selection criteria of five chosen feature attribution method is discussed and overview of them and their implementation are provided. In Section 4, a brief overview of chosen evaluation metrics is given and selection criteria for these metrics are discussed. In Section 5, the experimental setup which comprises the used network, data set, and metrics are explained. In Section 6 and 7, results of the experiment (comparison of five current-state-of-the-art methods using different evaluation metrics) is provided and discussed respectively. Lastly, in Section 8, a summary of the paper and future work to do are presented.

2 Related Work

Pixel attribution methods detect and highlight the relevant pixels for an image classification pixels by a neural network. These methods can be found in different names such as feature relevance, feature attribution, feature contribution, sensitivity map, saliency map, and pixel attribution map [22]. Pixel attribution can be categorized as a special case of feature attribution for images. Feature attributions mainly explain predictions by attributing each input feature measuring the change (negatively or positively) in the prediction. LIME [24], SHAP [21] and Shapley Values [11] are some of the examples of commonly used feature attribution methods.

Neural networks are considered that output as prediction a vector of length C, which consists of regression where $C = 1$. The output of the neural network I is $S(I) = [S_1(I), ..., S_C(I)]$ for an image. All these methods take $x \in \mathbb{R}^f$ with f features as an input, where x can be words, image pixels, tabular data etc., and output a relevance value for each f input features as explanation: $R^c = [R^c_1, ..., R^c_f]$. The c represents the relevance for the c-th output $S_C(I)$ [22].

Following represents the two different types of attribution methods:

- **Occlusion- or perturbation-based:** Manipulate parts of an image to provide explanations (model-agnostic). SHAP and LIME are the example methods of this category.
- **Gradient-based:** Compute the gradient of the prediction regarding the input features. In general, they differ in how the gradient is calculated.

In both approaches that are mentioned above, the explanation and the input image have the same size, and both assign a value to each pixel that can be explained as the prediction relevance of the pixel/classification of the image [22].

Pixel attribution methods can also be categorized with respect to the baseline equation:

- **Gradient-only methods:** Gradient-only attribution methods can be interpreted as follows: if a pixel were to be changed, the probability of the predicted class would go up (for positive gradient) or down (for negative gradient). A larger absolute value of the gradient means that the effect of a change at this pixel is stronger. They basically tell whether a change in a pixel would change the prediction [22]. Vanilla Gradient [30] and Grad-CAM [28] are two examples of gradient-only methods.

- **Path-attribution methods:** These methods compare the actual image to a reference image, which is a completely grey image, in other words artificial "zero", and the difference between the actual and the baseline prediction is distributed among the pixels. Model-specific gradient-based methods like DeepLift [29] and Integrated Gradients [32], and model-agnostic methods like LIME and SHAP belong to this category. Some path attribution methods such as SHAP and LIME are "complete", meaning that for all input features, the sum of the relevance values is the prediction of the image minus the prediction of a reference image. Path-attribution methods always make the interpretation with respect to the baseline, the difference between classification scores of the baseline and the actual image is assigned to the pixels. The choice of the reference image can affect the explanation significantly. Usually, a "neutral" image is used [22].

2.1 XAI methods

2.1.1 Integrated Gradients

Integrated gradients (IG) [32] aims to compute the importance value for each of the input features by considering gradients of the output with respect to the input of a machine learning model. Particularly, it represents the attribution for each of the input features by calculating the integral of the gradients taken along the path from a baseline instance to an input instance. Integral of the gradients are approximated using a Gauss Legendre quadrature rule [34] or a Riemann Sum [35].

2.1.2 DeepLift (Deep Learning Important FeaTures)

DeepLIFT [29] method is used for decomposing the output of the prediction on a given input by backpropagating the contributions of all neurons in a neural network to every input feature. It compares each neuron activation to its reference activation. Then, it assigns contribution scores according to the difference. In a single backward pass, each contribution score can be calculated efficiently.

2.1.3 Shapley value sampling

Shapley value [11] is one of the attribution methods which is based on a cooperative game theory concept. The method takes permutation of each input feature and adds them individually to the baseline. After adding each feature, the output difference represents the contribution. In order to obtain the attribution, all these differences are averaged over all the permutations.

2.1.4 LIME

LIME [24] is one of the interpretability methods that trains interpretable, surrogate models by sampling data points around an input example using evaluations of the model at these points in order to train simple, interpretable, surrogate model like a linear model.

2.1.5 Occlusion

Occlusion [39] is a perturbation-based method to compute attribution. It replaces each contiguous rectangular region with a given baseline and calculates the output difference. To compute the attribution for a specific feature, corresponding differences in output are averaged for the features located in multiple regions (hyper-rectangles). The occlusion method is commonly useful for the cases related to images, where the pixels are likely to be highly correlated in a contiguous rectangular region.

2.2 Evaluation metrics

2.2.1 (In)fidelity

Fidelity can be measured in the presence of apriori information that only a particular subset of features is relevant. In this case, the approach is to test whether the features with high explanation weights belong to this relevant subset [12]. More quantitative can be considered such as measuring the correlation between the difference in function value and the sum of a feature importances' subset, where such apriori information is absent [7]. A single fidelity measure called "infidelity" [38], which generalizes what is mentioned previously, is proposed. The proposed infidelity measure is defined as the expected difference between two terms: 1) the dot product of the input perturbation to the explanation and 2) the output perturbation, which is the difference in function values when significant perturbations are applied to the input.

2.2.2 Max-Sensitivity

Max-sensitivity [38] computes the maximum change in the explanation when a small perturbation is applied to the input x. It is always finite since the score of the explanation is bounded, and hence is more robust to estimate.

2.2.3 ROAR (RemOve And Retrain)

ROAR [17] is used to evaluate the approximate accuracy of feature attribution methods. It estimates the importance of an input feature in deep neural networks. It removes the fraction of most important (according to each estimator) input features. Then, it computes

the change of the model accuracy after retraining the model. The most important inputs, whose removal causes a gradual drop in the model performance compared to all other estimators, are identified by the most accurate estimator.

2.2.4 Pointing Game

The Pointing Game [40] determines the quality of a feature attribution method. It tests how well the attribution method can extract a response, which is correlated with the existence of given object categories in the image, from a predictor. For instance, given an input image that contains an object that belongs to category c, the attribution method can be applied to the predictor to identify the part of the images responsible for predicting category c. Generally, feature attribution methods return a saliency map. The saliency map should be converted into a single point that is most likely to be contained by an object of that class. If the point is within a tolerance value to the image region which contains the object, the attribution method scores a hit.

2.2.5 AUC-ROC

AUC-ROC [10] measures the performance of classification problems with various threshold settings. Receiver Operating Characteristic (ROC) is a probability curve, and Area Under the Curve (AUC) is the two-dimensional area underneath the ROC curve, which represents the measure of separability. It demonstrates the capability of a model to distinguish between classes. Higher AUC means that model is better at predicting the correct classes.

3 Methods

3.1 Selection criteria

As a five current state-of-the-art feature attribution methods, **Vanilla Gradients, SmoothGrad, XRAI, Grad-CAM** and **BlurIG** are chosen and will be compared. Decision is made according to below criterias that each method has in common:

- Recently developed,
- Outperforms commonly used methods (e.g. **Integrated Gradients** and **LIME**),
- Gradient-based image classification methods (usually faster to compute).

Comparison of chosen five current state-of-the-art methods does, to the best of my knowledge, not exist. In this section, we aim to provide a brief overview of these methods and how they work based on their algorithms.

3.2 XAI Methods

The remaining sections of XAI methods argue about the pros and cons of the different methods based on the papers that are related to each of the methods.

3.2.1 Vanilla Gradients

Vanilla Gradient [30] is one of the first pixel attribution approaches which is quite similar to the back-propagation. The gradient of the loss function is calculated for the interested class with respect to the input pixels. This provides a map of the size of the input features with negative to positive values. The algorithm works as follows:

- Perform a forward pass of the image of interest.
- Compute the gradient of a class score of interest with respect to the input pixels, and set all other classes to zero:

$$E_{grad}(I_0) = \frac{\partial S_c}{\partial I}|_{I=I_0}. \tag{1}$$

- Visualize the gradients either showing the absolute values or highlighting negative and positive contributions separately.

In Equation 1, I represents the image and $S_c(I)$ represents the score for class c.

Authors of [29] argues that Vanilla Gradient has a saturation problem. In the case of using ReLU, if activation goes under zero, activation will be capped at zero and will not change: e.g., let us assume the input to the layer is two neurons with weight -1 and -1 respectively, with the bias of 1. The activation will be the sum of the two neurons if the sum of them is less than 1 when passing through the ReLU layer. Activation will remain saturated at 1 if sums of the neurons are greater than 1. The gradient at this point will become zero, hence, Vanilla Gradient will conclude this neuron is not important [22].

3.2.2 SmoothGrad

SmoothGrad [31] is a feature attribution method that tends to decrease visual noise and it can be combined with other sensitivity map algorithms as well. The core of the algorithm is based on sampling from the image simply adding noise to it, then taking the average of sensitivity maps result for each image that is sampled. Additionally, the related paper points out that on sensitivity maps, there is an additional "de-noising" effect of adding noise (which is one of the common regularization techniques) at training time [9]. According to the paper, inferring with noise and training with noise has a complementary effect that yields the best result when the two techniques are performed together [31].

The issue with using gradients while measuring the influence of an important feature is the possibility of saturation of the class activation function S_c [33] [19]. In other words, this feature might have a strong effect globally but with a small derivative locally. there are many approaches such as Layerwise Relevance Propagation (LRP) [8], DeepLift [29], and recently Integrated Gradients [32] which tried to identify this issue by estimating the global importance of each pixel instead of estimating local sensitivity [31].

The derivative of the S_c might fluctuate at a small scale in a very sharp way. This means that noise seen in the sensitivity map may appear because of the local variations in partial derivatives. Since the networks typically are based on ReLU activation functions, S_c will not be continuously differentiable. So, derivatives will not vary smoothly either [31].

In these fluctuations, the gradient of S_c will be less meaningful at any point compared to the local average of gradients. To improve sensitivity maps, visualization of the gradient can be based on smoothing of ∂S_c with Gaussian kernel instead of visualizing it directly on the gradient ∂S_c [31].

A simple stochastic approach can be computed since it is intractable to compute local average in high dimensional input space directly. Random samples can be taken from the

neighborhood of an input x and the average of the resulting sensitivity maps can be calculated. Equation 2, where n represents the number of samples, and $\mathcal{N}(0,\sigma^2)$ is the Gaussian noise with standard deviation σ, explains what is suggested above mathematically [31]:

$$\hat{M}_c(x) = \frac{1}{n}\sum_{1}^{n} M_c(x + \mathcal{N}(0,\sigma^2)). \tag{2}$$

3.2.3 XRAI (eXplanation with Ranked Area Integrals)

It is a region-based saliency method that is based on Integrated Gradients (IG) [32]. XRAI [18] method works as follows:

- First, segments the image, then tests the importance of each region iteratively.
- Second, based on attribution scores, combines smaller segments into larger segments.

Experiments show that using this strategy, XRAI outperforms existing saliency methods by offering tightly bounded saliency regions in high quality. It can be used with any Deep Neural Network (DNN) model if it is possible to use some similarity metrics to cluster the input features into segments [18].

Gradient-based methods measure the sensitivity of output of the model with respect to changes in each individual input feature by taking the partial derivative of input feature i. Partial derivatives do not show precisely whether a given input feature contributes to the predicted class, they barely reveal whether the changes in the input feature affect the model prediction. Thus, some of the input features might be irrelevant to the predicted class, but have a high attribution still [18].

XRAI identifies relevant regions to the predicted class by discarding irrelevant ones which address the issue mentioned above. It satisfies the ”Completeness” properties of IG [32], which states that the sum of all features attributions must be equal to the difference between the model output corresponds to the instance input x and the model output at the baseline x_0 as seen Equation 3 [18].

$$\sum_i A_i(x,x_0) = F(x) - F(x_0) \tag{3}$$

It suggests that regions contributing to the predicted class should have high positive attribution, regions that are irrelevant should have near-zero attribution, and regions correspond to the other classes should have negative attribution [18].

XRAI usually works best on natural images that contain multiple objects. It is also better to have higher-contrast images since the method generates region-based attributions which produce the most salient heatmap regions that are human-readable and smooth for image classification [5].

3.2.3.1 Segmentation

In the related paper, they used Felzenswalb's graph-based method. They used the "skimage" python package [15] for segmentation [18].

3.2.3.2 Attribution

Normally, IG uses a black image as a baseline, which reduces the attribution of dark input pixels although they might have significant importance. To prevent this, XRAI uses IG with black and white baselines. It is guaranteed that for any pixels in the image, sum of the weight term will be 1.0, which satisfies the $|x-1.0|+|x-0.0|=1.0,\ \forall x \in [0.0, 1.0]$, where 1.0 and 0.0 correspond to black and white baselines respectively, and x is the input pixel value. As a result, all pixels can have a chance to contribute equally to the attributions without depending on the distance from the baseline. This method produces consistent saliency maps [18].

3.2.3.3 Selecting regions

For selecting regions, XRAI satisfies the Sensitivity-N [7], where the sum of all attributions of an instance input is equal to the softmax value of the input minus the softmax value of the baseline. For example, between given two regions, one which sums to the more positive value is more important for the classifier. Considering this, XRAI adds the regions selectively that yield the maximum gain, which contains more positive values compared to others, in the total attributions per area. XRAI algorithm runs till acquiring the whole image as the mask or if there are no regions left to add. Trajectories of masks can be considered as a ranking of the regions according to their importance.

3.2.4 Grad-CAM (Gradient-weighted Class Activation Mapping)

Grad-CAM leverages from gradient information flowing into the last convolutional layer of Convolutional Neural Network (CNN) to assign importance values to each neuron. By highlighting the important regions in the image, it produces a localization map for image prediction. Grad-CAM can be used in any layer of deep networks to explain activations [28], but (usually) to the last convolutional. Equation 4 is the formula to calculate localization map:

$$L^c_{Grad-CAM} \in \mathbb{R}^{u \times v} = \underbrace{ReLU}_{\text{Pick positive values}} (\sum_k \alpha^c_k A^k) \tag{4}$$

where u represents width, v represents the height of the explanation, c is the class of interest, and A^k is the k feature maps in the last convolutional layer [22].

Grad-CAM technique works as follows:

- Forward propagate the input image through the convolutional neural network.
- Obtain the raw score for the class of interest, meaning the activation of the neuron before the softmax layer.
- Set all other class activations to zero.
- Backpropagate the gradient of the class of interest to the last convolutional layer before the fully connected layers:

$$\frac{\partial y^c}{\partial A^k}. \tag{5}$$

- Weight each feature map pixel by the gradient for the class, where i and j refer to the width and height respectively. It means that gradients are pooled globally:

$$a^c_k = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backpropagation}} . \tag{6}$$

- Calculate an average of the feature maps, weighted per pixel by the gradient.
- Apply ReLU to the averaged feature map.
- For visualization: Scale values to the interval between 0 and 1. Upscale the image and overlay it over the original image [22].

3.2.5 BlurIG (Blur Integrated Gradients)

BlurIG [37] has several advantages when compared with the other techniques:

- First, It can identify at what scale the network recognizes the object. Paper points out that it produces scores in scale/frequency dimension that captures interesting phenomena.

- Second, it holds the scale/space axiom [20], which employs perturbations that are artifact-free. BlurIG can produce consistent and clear results when it is applied to deep networks.
- Third, for perception task, it annihilate the necessity of baseline parameter for IG [32], which is desirable since the choice of baseline might have a significant effect on explanations.

Previous attribution methods do not produce localization in frequency. They produced feature importance for pixels, in other words, points in space. BlurIG proposes a new approach to produce explanations in both scale/space frequency and space, which means that it can tell whether the detection is based on large-scale feature or fine-grained feature [37].

All attribution techniques include perturbations either of the networks state [30] or the inputs [32]. Removing an important feature leads to a significant change in the prediction score. Paper stated that no discussion exists in the literature whether perturbations can add features unintentionally. The change in the prediction score can be the result of the detection of the different objects and not a result of the information destruction. This leads to the explanation of influential features that do not exist in the input actually. Paper discusses how the scale/space theory points out accidental feature creation [37].

4 Evaluation Metrics

4.1 Selection criteria

As evaluation metrics, **MoRF, Smoothness, Similarity to Edge Map, Invariance to Model Weights** and **Invariance to Data Labeling** are chosen and will be used for comparison of five current-state-of-the-art methods. The decision is made according to below criteria that each method has in common:

- Implementing the metrics is doable,
- Established metrics,
- Can be computed efficiently.

4.2 MoRF (Most Relevant First)

The intuition regarding the MoRF, in other words "deletion", metric is that a change in the decision will be forced upon the base model due to the 'cause' having been removed. The probability of the predicted class is measured by the metric, specifically, the decrease in probability when important pixels keep getting removed. The importance of pixels is stated in the importance map. A sharp drop leading to an area low under the probability curve (as a function of the proportion of pixels removed) implies a good explanation [23] [27]. According to Merijn Schroder's presentation, MoRF outperforms namely ROAR [17], Cropping [13], Max-Sensitivity [38], and (In)fidelity [38] metrics. Therefore, we decided to use this metric for the evaluation of feature attribution methods.

4.3 Smoothness

The smoothness metric is, to the best of my knowledge, a novel XAI explanation evaluation metric. Although it has been proposed before to use the metric in other computer vision applications, we believe it has not been applied before in the context of XAI method explanation evaluation. The metric involves computing the Laplacian (gradient) of an explanation image *e(x,y)*. This is obtained through the convolution $g(x,y) = \omega * e(x,y)$, where $*$ denotes the convolution operator, and ω is the 2D Laplacian kernel

$$\omega = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{7}$$

The resulting image $g(x,y)$ highlights each pixel (x,y) in the explanation of how different it is from the pixels which are neighbors of the pixel: e.g., when a pixel is considered just as important as the pixels which are next to it, it will have a gradient of 0. Then, the absolute value of each pixel is computed, i.e. $h(x,y) = ||g(x,y)||$. Lastly, the average of all the pixels is computed resulting in the average absolute gradient of the explanation: i.e., the smoothness of the explanation

$$Smoothness(e) = \#\text{dom}(e)^{-1} \times \sum_{(x,y) \in \text{dom}(e)} || \omega * e(x,y) ||. \tag{8}$$

When this value is large, the explanation is not smooth. This is arguably worse, as the explanation is noisier than needed. When the value is smaller, this is better, as the explanation does not contain more noise than needed.

4.4 Invariance to Model Weights

The output of an XAI method is not necessarily guided by the weights of the model it tries to explain. An approach to evaluate to what degree an XAI method is invariant to model weights was proposed in [6]. The authors compared the output of an XAI method on a trained model with the output of the XAI method on a model with the same architecture, but with random weights. We use the same approach to determine invariance to model weights. In my experiments, the class label for which evidence is explained is kept constant, even when the model classifies input differently due to randomization of weights.

4.4.1 Cascading Randomization

Similar to [6], we determined how explanations change when we set the weights of layers/blocks (from top to bottom) iteratively randomly. In iteration i, the weights of the top layer/blocks until the ith top layer is set randomly.

We defined the cascading randomization invariance to model weights metric as

$$CascadingRandomization(e) = spearmanr(\ e_f(x,y), e_{fcr_i}(x,y)), \tag{9}$$

where $e_f(x,y)$ is an explanation of trained network $f(x,y)$, and $e_{fcr_i}(x,y)$ is an explanation of network $fcr_i(x,y)$. $fcr_i(x,y)$ is equal to $f(x,y)$ except the layers/blocks from the top layer until the ith layer counted from the top layer are set with random weights.

4.4.2 Independent Randomization

Similar to [6], we determined how explanations change when we set the weights of layers/blocks (from top to bottom) iteratively randomly. In iteration i, the weights of the ith top layer is set randomly: i.e., the layer that is i layers/blocks away from the top layer.
We define the independent randomization invariance to model weights metric as

$$IndependentRandomization(e) = spearmanr(\ e_f(x,y), e_{fr_i}(x,y)), \tag{10}$$

where $e_f(x,y)$ is an explanation of trained network $f(x,y)$, and $e_{fr_i}(x,y)$ is an explanation of network $fr_i(x,y)$. $fr_i(x,y)$ is equal to $f(x,y)$ except the ith layer counted from the top layer is set with random weights.

4.5 Invariance to Data Labeling

The authors of [6] proposed to compare explanations of XAI methods of a model trained on a data set with those of a model trained on the same data set, but with the labels randomly permuted. The difference between explanations of both models should give insight into the degree to which the XAI method is invariant to the input data.

Due to time constraints, we were unable to re-train the Inception-v3 model which we used during the experiments. Instead, we took a slightly different approach than the authors of [6]. We determined the difference between an explanation for the predicted label with that for a random other label. Obviously, this limits the significance of the results: e.g. because the choice of the random label is arbitrary. The difference between explanation pairs was expressed in terms of their Spearman Rank correlation [1].

4.6 Similarity to Edge Map

The authors of [6] found that XAI methods that are insensitive to model weights and input data commonly produce explanation images that are strikingly similar to edge maps of the input image. Since correlation does not necessarily imply causation, it cannot be concluded from this that explanations that are similar to edge maps inherently have little informative power. Nonetheless, we will try to observe whether methods that score poorly on this metric also typically score poorly on other metrics. We defined the similarity to edge map metric as

$$EdgemapSimilarity(e) = spearmanr(e(x,y), ||\ \omega * f(x,y)\ ||), \tag{11}$$

where ω is the 2D Laplacian kernel of Equation 7, $f(x,y)$ is the explained network, $e(x,y)$ is the explanation, and $*$ denotes the convolution operator. $spearmanr(x_1,x_2)$ denotes the

Spearman Rank correlation between x_1 and x_2. Spearman Rank correlation coefficient [36] ρ is

$$\rho = 1 - \frac{6\sum(d_i^2)}{n(n^2-1)}, \tag{12}$$

where n denotes the number of observations, and d_i denotes the difference between the two ranks of each observation

$$d_i = rg(x_1) - rg(x_2). \tag{13}$$

Spearman Rank correlation results in 1 or -1 when the two variables have a monotonic relationship even if the relationship is not linear: e.g., if the value of one variable increases, the other one also increases, but the amount is not consistent. If the result of the Spearman test is 1, there is a perfect association of ranks. If it is -1, there is a perfect negative association of ranks. If the result is 0, it means there is no relationship between two variables, hence, no association of ranks [36].

5 Experimental Design

5.1 Network

For the experimentation, a pre-trained Inception-v3 convolutional neural network model is used. It is a widely-used image recognition model that has been shown to attain approximately 82.8% accuracy on top-1 prediction with an ensemble of 4 models and multi-crop evaluation on the ImageNet data set in the 2012 ILSVR (ImageNet Large Scale Visual Recognition Challenge) [33]. As a result, the network provides good insight for explainable AI methods, thus, there is no need to evaluate other networks since Inception-v3 already has a high prediction accuracy. To see the network architecture, please refer to [33].

5.2 Data set

For the experimentation, ImageNet data set is used to train the Inception-v3 network. It is a large data set of over 14 million images. It has more than 20,000 categories. It was designed by academics intended for computer vision research and is available for free to researchers for non-commercial use [26]. As a test set, ImageNet validation set is used. It contains 1000 categories (classes), including over 3000 images. For demonstration purposes, five of them (doberman, volcano, fig, bullmastiff, and goldfish) are presented in the paper. The other two images, pug and pug with glasses, that are presented in the paper are taken from [2] and [3] respectively.

5.3 Metrics

For the experimentation, MoRF, Smoothness, Similarity to Edge Map, Invariance to Model Weights, and Invariance to Data Labeling evaluation metrics are used to compare five current state-of-the-art methods. A detailed description of each evaluation metrics mentioned above is provided in Section 4.

6 Results

Results for saliency maps and MoRF are generated with the help of the PAIR-saliency[1] and eclique-RISE[2] repositories respectively. Implementation can be found in RUG-research-internship[3] repository which includes all parameter setting choices in code comments.

6.1 Quantitative Results

In Figure 1, visualization of the evaluation metric results for seven feature attribution methods applied to seven test set images that are presented in Table 1 can be found in terms of MoRF (deletion), smoothness, and log_{10} scale of time.

Figure 2 represents a comparison of a sample distribution across different feature attribution methods. The first plot represents the distribution of MoRF of each method, the second one smoothness of the methods, and the third one average of log_{10} scale of time to run each method.

Table 1 shows the average results of four evaluation metrics namely MoRF, smoothness, and the runtime in seconds on seven images from the test set. Grad-CAM$_S$ and BlurIG$_S$ represent the smoothed versions of Grad-CAM and BlurIG methods respectively.

Due to time constraints, some of the entries are not available for specific methods since running some of the methods (especially for BlurIG$_S$) and computing some of the metric results for specific methods take enormous time. Because of the same reason, we keep our sample size restricted with seven images for the methods, and even less for some of the evaluation metric computations. The similarity to edge map and invariance to data metrics could only be computed on a single random test set item. Due to time constraints, we were not able to randomize all layers/blocks of the model in invariance to model weights experiments again due to the computational time complexity. The test set would have been included more sample images, and all metrics would have been computed on the entire test set if we had more time.

[1] https://github.com/PAIR-code/saliency

[2] https://github.com/eclique/RISE

[3] https://github.com/aycavci/RUG-research-internship

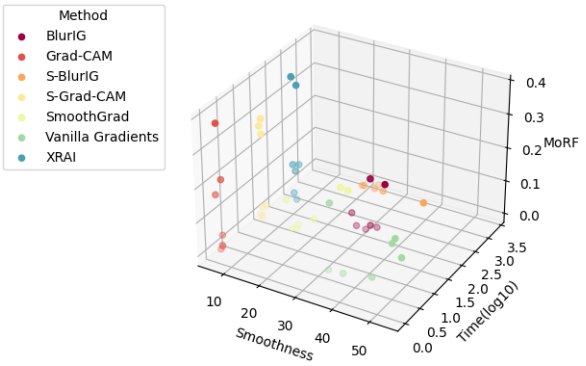

Figure 1: Visualization of results presented in Table 1.

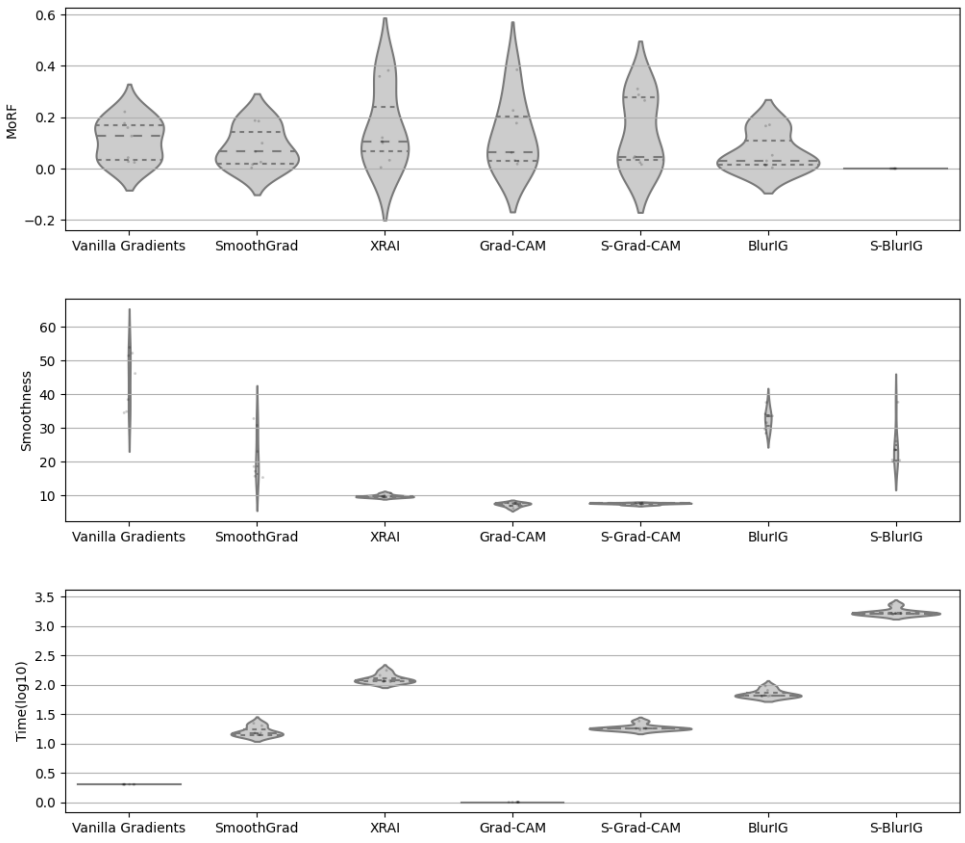

Figure 2: Visualization of results presented in Table 1.

Method	MoRF/Deletion	Smoothness	Similarity to Edge Map	Invariance to data	Runtime
Vanilla Gradient	0.111 ± 0.074	41.510 ± 7.814	0.10	0.86	2
SmoothGrad	0.083 ± 0.072	20.559 ± 6.698	0.15	0.79	15.286
XRAI	0.158 ± 0.140	9.776 ± 0.396	-	-	118
Grad-CAM	0.133 ± 0.127	7.370 ± 0.558	0.13	-	1
Grad-CAM$_S$	0.142 ± 0.127	7.599 ± 0.216	0.13	-	18.714
BlurIG	0.064 ± 0.067	31.687 ± 2.836	0.25	0.43	66.429
BlurIG$_S$	-	24.553 ± 5.727	0.17	-	1728.857

Table 1: **MoRF (Deletion)**: Mean $\pm$ STD AUC results of MoRF (Deletion) metric. Smaller value means better explanation. **Smoothness**: Mean $\pm$ STD Smoothness of each explanation expressed in terms of the average of all the pixels resulting average absolute gradient of the explanation. Lower value indicates that explanation is smooth, meaning that it does not contain more noise than needed. **Similarity to Edge Map**: Similarity between edge map of original images and explanations expressed in terms of Spearman Rank correlation. Smaller value means better explanation. **Invariance to data**: Invariance to data labeling is expressed in the Spearman Rank correlation between an explanation for the true label and an explanation for a random label. Lower value means better explanation. **Runtime**: Average runtime in seconds to create a single explanation. All metric results are averaged over seven images from the test set. Entries with "-" are unavailable.

6.1.1 Invariance to Model Weights

6.1.1.1 Cascading Randomization

In Figure 3, three feature attribution methods namely Vanilla Gradients, SmoothGrad, and BlurIG compared in terms of their Spearman Rank correlation between the explanation of the original model, and explanation of model with all layers between the top layer and the randomized layer indicated on the x-axis. High Spearman Rank correlation difference between fully connected (fc) layer and AuxLogits layer indicates better explanation.

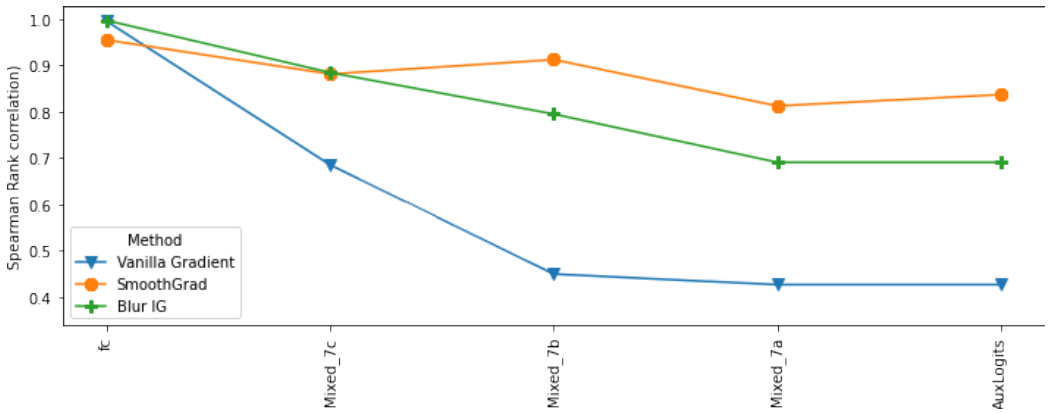

Figure 3: Spearman Rank correlation between explanation of original model, and explanation of model with all layers between the top layer and the on the x-axis indicated layer randomized. Inspired by Figure in [6].

6.1.1.2 Independent Randomization

In Figure 4, three feature attribution methods namely Vanilla Gradients, SmoothGrad, and BlurIG compared in terms of their Spearman Rank correlation between the explanation of the original model, and explanation of model with the randomized layer indicated on the x-axis. High Spearman Rank correlation difference between fully connected (fc) layer and AuxLogits layer indicates better explanation.

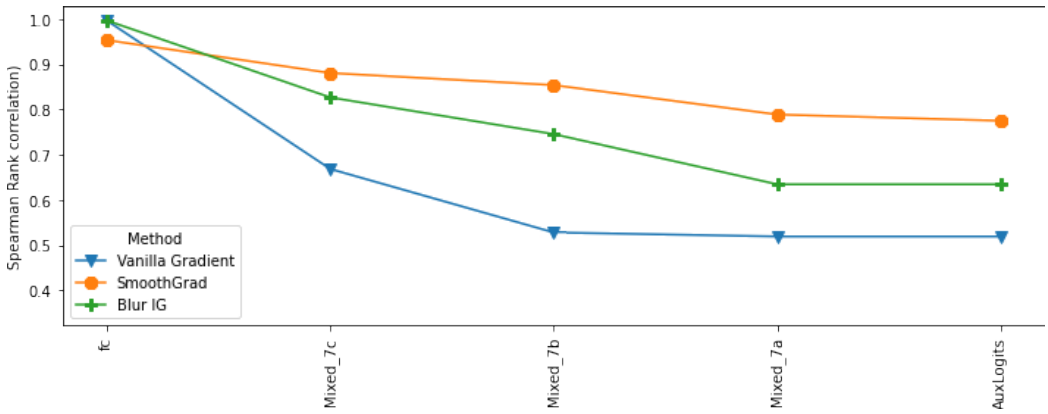

Figure 4: Spearman Rank correlation between explanation of original model, and explanation of model with the on the x-axis indicated layer randomized. Inspired by Figure in [6].

6.2 Qualitative Results

6.2.1 Saliency Map examples

Figures 5-11 show saliency map examples for seven pictures that are chosen from the test set. The first six images are correctly labeled by the network, and the last one is labeled as "sunglasses" instead of "pug". In grayscale saliency maps, white parts highlight the most important features to predict the images. In colored saliency maps, red parts highlight the most important features to identify the images correctly.

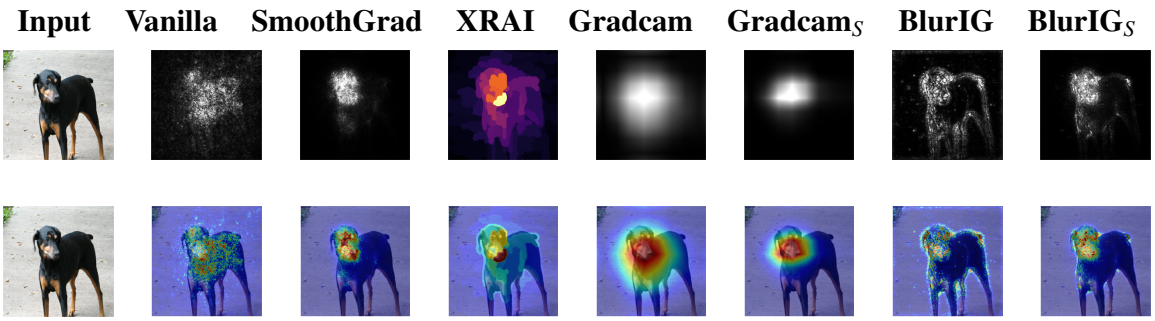

Figure 5: Saliency maps for doberman.

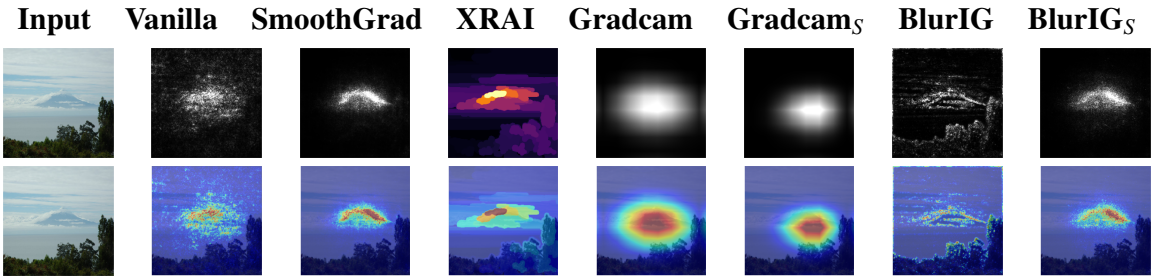

Figure 6: Saliency maps for volcano.

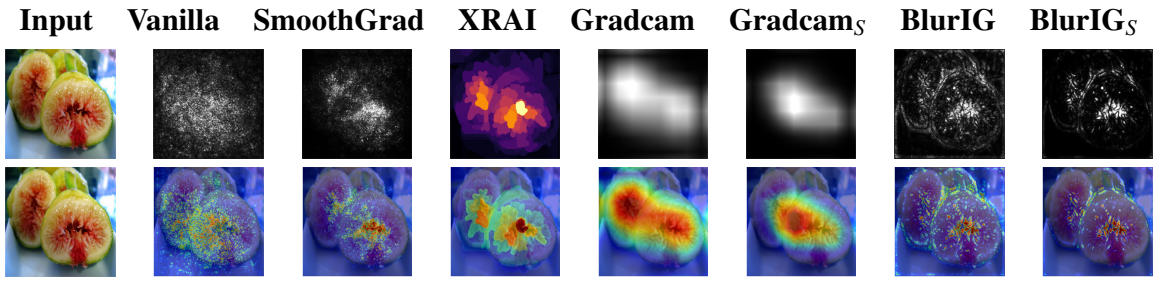

Figure 7: Saliency maps for fig.

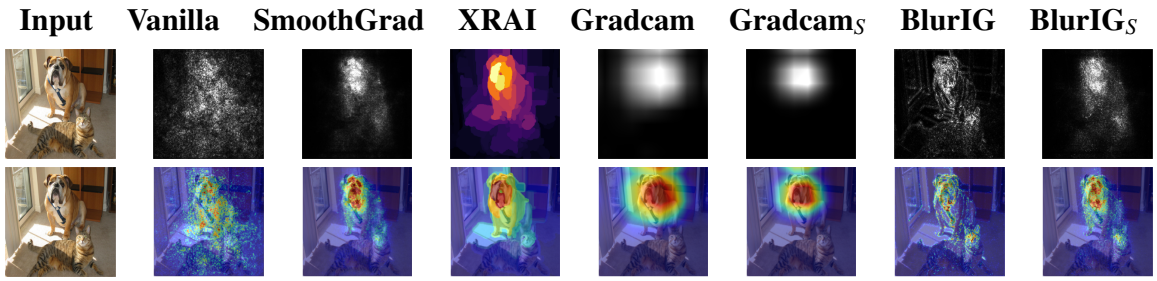

Figure 8: Saliency maps for bullmastiff.

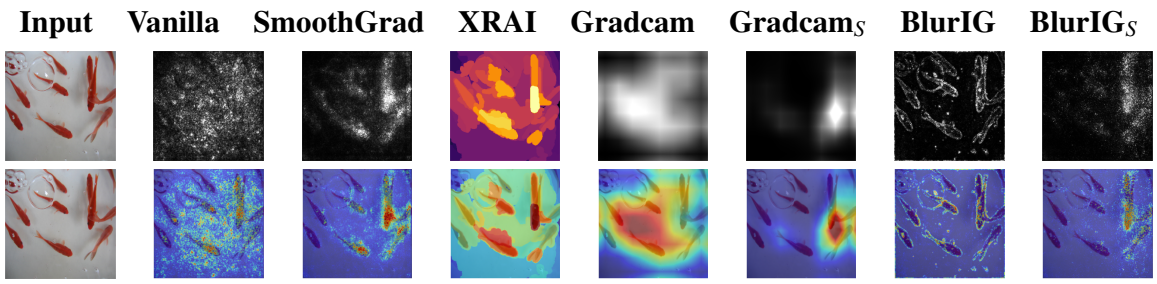

Figure 9: Saliency maps for goldfish.

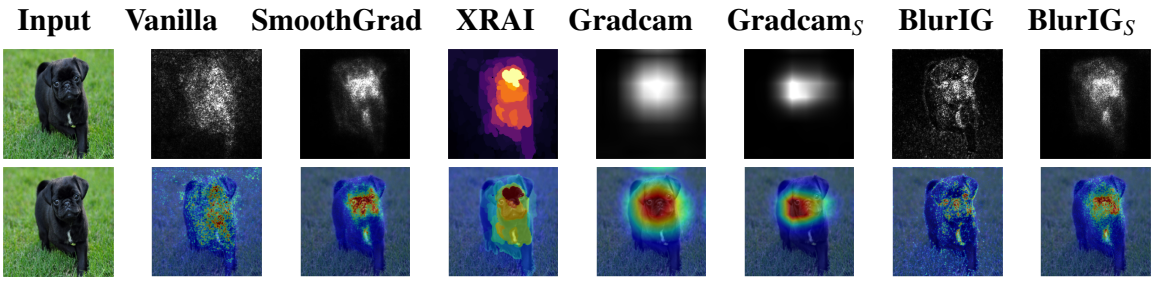

Figure 10: Saliency maps for pug.

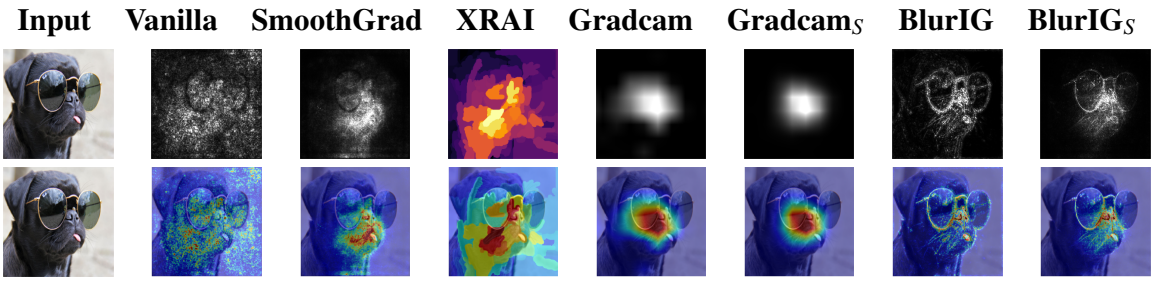

Figure 11: Saliency maps for pug with glasses.

6.2.2 Invariance to Model Weights

6.2.2.1 Cascading Randomization

In Figure 12, saliency maps for bullmastiff image are generated for three feature attribution methods namely Vanilla Gradient, Smooth Gradient, and BlurIG after cascading randomization is applied. The low similarity between images for each method itself means a better explanation the method has.

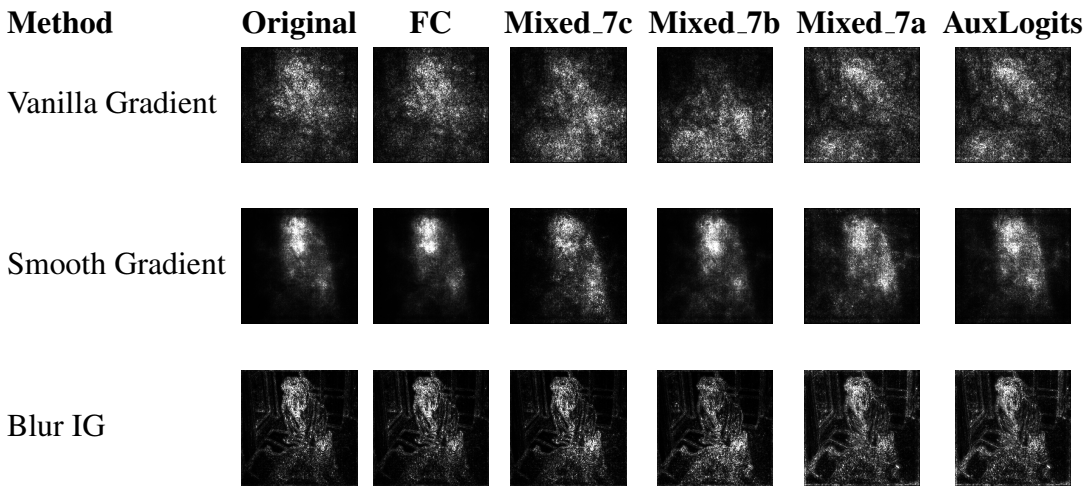

Figure 12: The explanation of the original model, and the explanation of the model with all layers between the top layer and the on the x-axis indicated layer randomized. Inspired by Figure in [6].

6.2.2.2 Independent Randomization

In Figure 13, saliency maps for bullmastiff image are generated for three feature attribution methods namely Vanilla Gradient, Smooth Gradient, and BlurIG after independent randomization is applied. The low similarity between images for each method itself means a better explanation the method has.

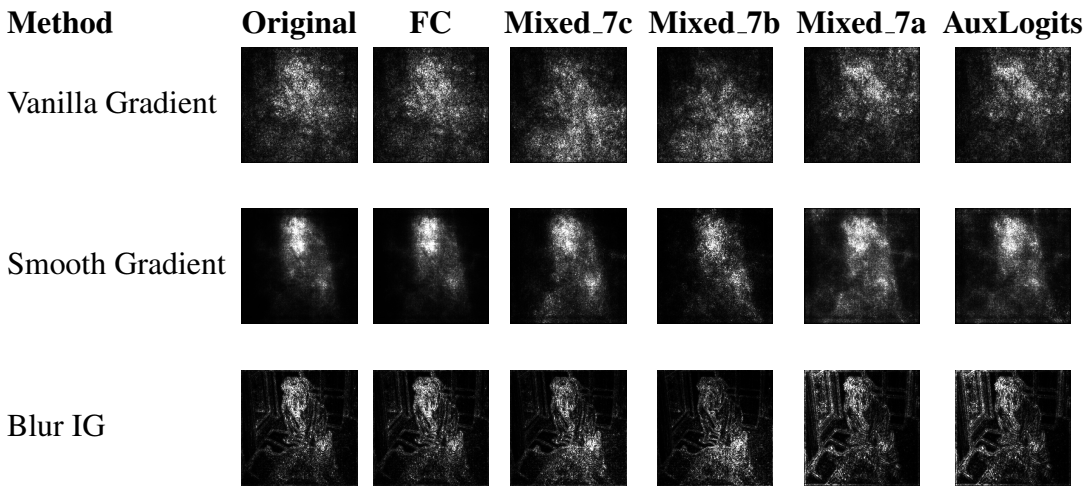

Figure 13: The explanation of the original model, and the explanation of the model with the on the x-axis indicated layer randomized. Inspired by Figure in [6].

6.2.3 Invariance to Data Labeling

Figure 14 represents a comparison between the explanation for a true label and for a random label of doberman, volcano, and fig images respectively. "True" columns show explanations for the true label and "Random" columns show explanations for a random label. If the similarity of the images between true and random labels for each method itself is low, the explanation of that method is better.

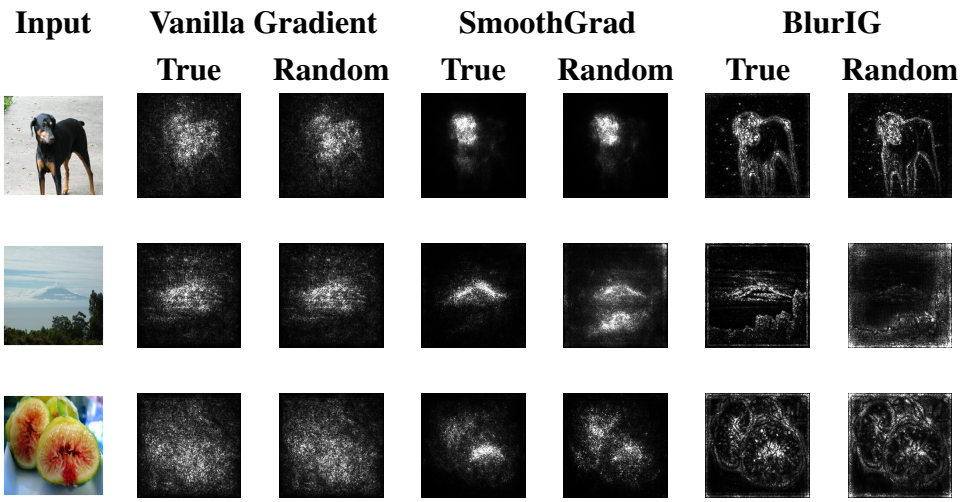

Figure 14: Comparison between the explanation for a true label and for a random label. "True" columns show explanations for the true label. "Random" columns show explanations for a random label. Inspired by [6].

6.2.4 Similarity to Edge Map

Figure 15 represents a comparison between the edge maps of original images of doberman, volcano, and fig, and their explanations for seven feature attribution methods. The low similarity between edge maps and the explanation of a specific method means that this specific method has a better explanation compared to others.

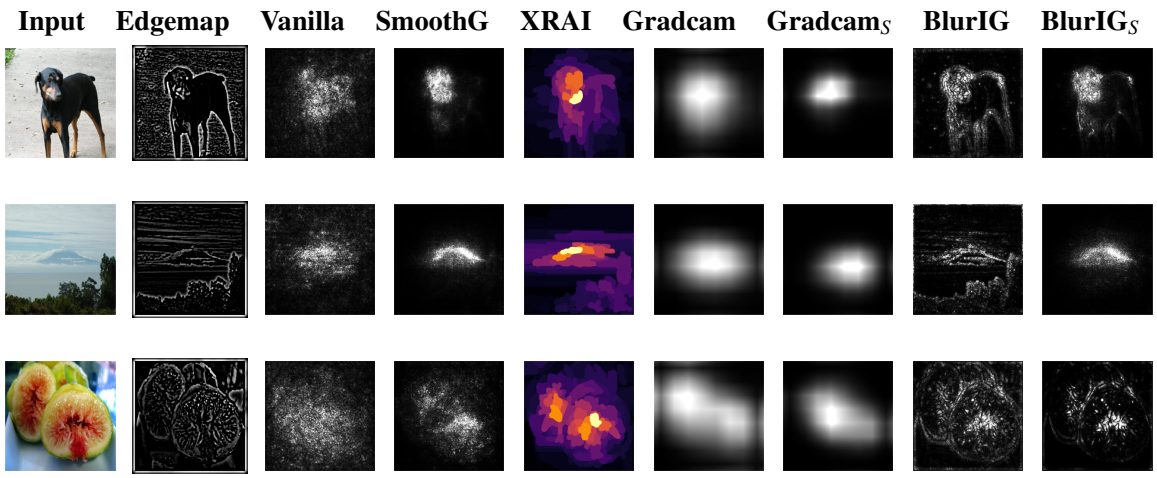

Figure 15: Comparison between the edge map of original images and the explanations. Inspired by Figure in [6].

7 Discussion

7.1 Quantitative Results

In Figure 1, we show a 3D visualization of evaluation metric results i.e., MoRF (deletion), smoothness and log_{10} scale of time. Seven feature attribution methods, which are listed under "Method" in the 3D graph, are applied to seven test set images. Grad-CAM$_S$ and BlurIG$_S$ represent the smoothed versions of Grad-CAM and BlurIG methods respectively. For a better explanation, what we expect is smaller MoRF and smoothness values. Runtime value does not indicate whether an explanation is better, but, if more than one method has very close MoRF and smoothness values, the runtime can be the decision-maker which method to prefer: e.g. if the MoRF and smoothness values of two methods are pretty close to each other, assume that they have almost equally good explanation, we can prefer the method has short runtime. Results show that Grad-CAM and Grad-CAM$_S$ are faster to compute compared to the others and has better explanation according to smoothness metric along with Grad-CAM$_S$. XRAI performs well, but slightly worse than Grad-CAM looking at the smoothness. BlurIG outperforms others according to the MoRF metric, but takes more time to compute when compared with other methods, except BlurIG$_S$, and has a better explanation than Vanilla Gradients when we look at smoothness. SmoothGrad has also a better explanation according to MoRF, but slightly worse than BlurIG still.

Figure 2 illustrates the metric value distributions of the by us evaluated feature attribution methods. The first plot represents the distribution of MoRF of each method, the second one smoothness of the methods, and the third one the average of log_{10} scale of time to run each method. The first plot shows that Vanilla Gradients, SmoothGrad, and BlurIG have a small standard deviation compared to the other methods namely XRAI, Grad-CAM, and Grad-CAM$_S$ (they all have high standard deviation), which indicates that the sample distribution is close to the mean. It points out that methods that have small standard deviation perform similarly on different images, which is considered as good meaning that method performance is image independent. Yet, since we only used seven test set images during the experimentation, it would be very naive to consider standard deviation as a reliable indicator. In the second plot, seven methods were compared using the smoothness metric. From the plot, we can observe that Grad-CAM and Grad-CAM$_S$ have a small mean value and standard deviation, and XRAI has a slightly higher mean and standard deviation compared to them. According to the smoothness metric, Vanilla Gradient performs poorly with the highest mean and standard deviation. The third plot visualizes the runtime of the methods in $\log_{10}$ scale. Grad-CAM and Vanilla Gradient have the smallest runtime around 1 and 2 seconds respectively. SmoothGrad and Grad-CAM$_S$ have the second smallest runtime values, and, BlurIG and XRAI follow them in the third and fourth place respectively. BlurIG$_S$ has the worst runtime with 29 minutes approximately.

Table 1 shows the mean quantitative results of five evaluation metrics namely MoRF, smoothness, similarity to edge map, invariance to data labeling, and the runtime in seconds on seven images from the test set. An explanation is considered better when the explanation is not similar to an edge map. If the invariance to data labeling value is low, the explanation is considered better. Results of these two metrics are discussed in Section 7.2. The other three metrics' results (i.e., MoRF, smoothness, and runtime) were already discussed in the previous paragraphs. As was argued in previous paragraphs, according to the MoRF metric BlurIG, and according to smoothness metrics Grad-CAM provides a better explanation with 0.064 ± 0.067 and 7.370 ± 0.558 respectively. Interestingly, for the smoothness metric, Grad-CAM$_S$ performs slightly worse than Grad-CAM, since we expect the other way around. Grad-CAM$_S$ scored 7.559 ± 0.216 on smoothness. The reason behind this peculiarity might be the usage of a small test set of (seven) images. We can see that even though Grad-CAM has a lower score, it has a high standard deviation, which means with more samples, it might have a higher average mean of smoothness than Grad-CAM$_S$ will have. According to similarity to edge map, Vanilla Gradient provides a better explanation with 0.1, and, BlurIG has the worst explanation with 0.25. On the contrary, invariance to data labeling metric argues that BlurIG has the best explanation among other methods with 0.43, and, Vanilla Gradient has the worst with 0.86. SmoothGrad and Vanilla Gradient methods are faster to compute with 1 and 2 seconds respectively, while BlurIG$_S$ is the slowest one with 1729 seconds (29 minutes) approximately.

In Figure 3 and 4, three feature attribution methods, namely Vanilla Gradients, SmoothGrad, and BlurIG are compared using the cascading randomization and independent randomization metrics respectively. Comparison is made for cascading randomization in terms of their Spearman rank correlation between an explanation of the original model, and explanation of a model with all layers between the top layer and the layer indicated on the x-axis randomized, and, for independent randomization in terms of their Spearman rank correlation between an explanation of the original model, and explanation of a model with the on the x-axis indicated layer randomized. When we both look at the Figures 3 and 4, we can observe that for Vanilla Gradient, the Spearman Rank correlation difference is 0.55 approximately between the AuxLogits layer and fully connected (fc) layer. It is 0.35 and 0.11 approximately for BlurIG and SmoothGrad methods respectively. Since the high Spearman rank correlation difference between AuxLogits and fc layer indicates better explanation, Vanilla Gradients outperforms SmoothGrad, and slightly BlurIG.

7.2 Qualitative Results

Figures 5-11 show saliency map examples for seven pictures that are chosen from the test set. The first six images are correctly labeled by the network, and the last one is labeled as

"sunglasses" instead of "pug". In grayscale saliency maps, white parts highlight the most important features to predict the images. In colored saliency maps, red parts highlight the most important features to identify the images correctly. In Figures 5, 6, 7, 8 and 9, Vanilla gradient is too noisy. Explanations of Grad-CAM have low resolution and they are blurry because the method uses the feature maps produced by the last convolutional layer of a network, which does not contain fine details. Explanations of BlurIG are very similar to the edge maps whereas explanations of BlurIG$_S$ differ substantially from the edge maps. For Figur e 5, 6 and 8, explanations of SmoothGrad and BlurIG$_S$ looks similar. In XRAI method, red highlighted parts are similar to the red parts in explanations of SmoothGrad especially for Figures 7, 8 and 9, which refers to the most important features for prediction. In Figure 10 and 11, we visualized saliency maps for two pug images, the latter one with the glasses. Trained network predicted Figure 11 as "sunglasses" instead of "pug". Interestingly, when these two figures are compared, we saw that some of the explanation methods, specifically SmoothGrad, Grad-CAM, Grad-CAM$_S$, and BlurIG$_S$ highlight the same part (nose area of the pug) in both images even though one of latter one is classified wrong. Additionally, none of the methods highlighted the sunglasses except BlurIG and Vanilla Gradient. This observation is quite interesting because we would expect that explanation methods should be model-dependent and expect them to explain the decision of the model. We do not know sure whether some of the methods are model-independent or the model is not being accurate for that specific image. For further investigation, more images that are classified wrong should be used during the experimentation to see whether the explanations actually change or not for feature attribution methods.

In Figure 12 and 13, we show explanations of the original model, and respectively explanations of models with all layers between the top layer and the on the x-axis indicated layer randomized or explanations of models with the on the x-axis indicated layer randomized. Explanations were generated by Vanilla Gradients, SmoothGrad, and BlurIG on the "bull-mastiff" image. An explainable AI (XAI) method should explain the weights of a model. When we make the weights of the model random, the explanations should change. If they stay the same, it means that the XAI method does not explain the weights of a model. If the explanations change, it means that explanations are model-dependent as they should be. We observed that, for both metrics, when the lower layer convolutional weights are randomized Vanilla Gradient and SmoothGrad masks change significantly, which means gradients show sensitivity to weight changes For BlurIG, the mask also changes, however, the resulting mask (original) is dominated by input structure still, which means it is invariant to layer weight changes, especially for higher-level layers. As a result, Vanilla Gradient and SmoothGrad are more sensitive to model weights, on the other hand, masks derived from BlurIG remain similar visually to masks of the trained model. Just looking at the masks, it is hard to say which method, Vanilla Gradient or SmoothGrad, actually provides a better explanation due to human perception.

In Figure 14, we show a comparison between explanations for the true label and a random label assigned to the images of doberman, volcano, and fig for Vanilla Gradient, SmoothGrad, and BlurIG methods. When we change the label, we change the sub-network that is explained. As a result, the explanations should change as well. Imagine we have a picture with a dog with glasses on. When we request an explanation for the label "dog", the dog should be highlighted by the XAI method. However, when we change the label to the class "glasses", the glasses should be highlighted instead. If both of these explanations are the same, it means the XAI method is independent of the model, and does not explain anything about the relationship between the image and the correct class that it belongs to, which it should not be. Results show that the explanation of SmoothGrad undergoes significant changes for images of volcano and fig, whereas it only changes slightly for the image of doberman. For BlurIG, explanation changes substantially for the image of volcano, but not for the images of doberman and fig. Vanilla Gradient seems that it has invariance to data labeling since the explanations do not differ significantly between true and randomly labeled images. Therefore, it is hard to say which method provides a better explanation due to human perception since explanations changes for some images while not changing other ones for a specific method, hence, the decision can be biased.

Edge detectors are one of the most classical tools to point out sharp transitions on images. Typically, they are not trained and they do not depend on predictive models.
Humans tend to interpret explanations that highlight edges in the input image as a good explanation, even though the explanations can be nonsensical (e.g. generated by an input image edge detector) [6]. It was found in [6] that explanations that are of low quality typically look like edge maps. However, this is not necessarily the case. Hence, it is not sufficient for visual inspection to differentiate edge detectors from explanations that are model-sensitive and conclude from this how good the explanations of an XAI model are.
In Figure 15, we compared an edge map of original images doberman, volcano, and fig, and their explanations for the feature attribution methods. We observed that the saliency explanations of BlurIG mostly capture the edges in the input images. This is in contrast with Vanilla Gradient and Grad-CAM that do not highlight edges in the original image.

8 Conclusion

In this research, we compared five state-of-the-art feature attribution methods (i.e., Vanilla Gradients, SmoothGrad, XRAI, Grad-CAM, and BlurIG) by evaluating them using five evaluation metrics (i.e., MoRF, smoothness, similarity to edge map, invariance to model weights, and invariance to data labeling). The results showed that BlurIG has a better performance in terms of providing better explanation compared to other methods according to the MoRF and invariance to data labeling metrics. The smoothness metric suggests that Grad-CAM provides a better explanation, whereas the similarity to edge map and invariance to model weights metrics advocate that Vanilla Gradient presents a better explanation. The Vanilla Gradients and Grad-CAM methods are fast (it takes only a couple of seconds to generate an explanation), on the other hand, BlurIG$_S$ takes several minutes. Runtime is not correlated with the explanation quality, but might be useful when it comes to preference between methods that have similar explanations. To sum up, we concluded that it is currently not possible to explicitly say which method is best since the metric results do not agree on which method provides the best explanation.

8.1 Future Work

Experiments should be repeated with a larger test set, and additional evaluation metrics to obtain more accurate and reliable results.

Bibliography

[1] *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY, 2008.

[2] Black pug: The complete care guide. https://www.perfectdogbreeds.com/black-pug/, 2021. [Online; accessed 24-June-2021].

[3] Dog with glasses wallpapers. https://wallpaperaccess.com/dog-with-glasses, 2021. [Online; accessed 24-June-2021].

[4] Introduction to ai explanations for ai platform. https://cloud.google.com/ai-platform/prediction/docs/ai-explanations/overview, 2021. [Online; accessed 17-May-2021].

[5] Limitations of ai explanations; ai platform prediction; google cloud. https://cloud.google.com/ai-platform/prediction/docs/ai-explanations/limitations, 2021. [Online; accessed 17-May-2021].

[6] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

[7] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks, 2018.

[8] A. Binder, G. Montavon, S. Bach, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for neural networks with local renormalization layers, 2016.

[9] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.

[10] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[11] J. Castro, D. Gómez, and J. Tejada. Polynomial calculation of the shapley value based on sampling. *Computers and Operations Research*, 36(5):1726–1730, 2009. Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X).

[12] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[13] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers, 2017.

[14] L. Edwards and M. Veale. Slave to the algorithm? why a 'right to an explanation' is probably not the remedy you are looking for. *Duke law and technology review*, 16:18–84, 2017.

[15] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 09 2004.

[16] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. Xai—explainable artificial intelligence. *Science Robotics*, 4(37), 2019.

[17] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks, 2019.

[18] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry. Xrai: Better attributions through regions, 2019.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278 – 2324, 12 1998.

[20] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990.

[21] S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent individualized feature attribution for tree ensembles, 2019.

[22] C. Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

[23] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models, 2018.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016.

[25] T. Rieg, J. Frick, H. Baumgartl, and R. Buettner. Demonstration of the potential of white-box machine learning approaches to gain insights from cardiovascular disease electrocardiograms. *PLOS ONE*, 15(12):1–20, 12 2020.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[27] W. Samek, A. Binder, G. Montavon, S. Bach, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned, 2015.

[28] 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. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019.

[29] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences, 2019.

[30] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[31] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

[32] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *CoRR*, abs/1703.01365, 2017.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[34] Wikipedia contributors. Gauss–legendre quadrature — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Gauss%E2%80%93Legendre_quadrature&oldid=1012677738`, 2021. [Online; accessed 12-June-2021].

[35] Wikipedia contributors. Riemann sum — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Riemann_sum&oldid=1028317435`, 2021. [Online; accessed 12-June-2021].

[36] Wikipedia contributors. Spearman's rank correlation coefficient — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Spearman%27s_rank_correlation_coefficient&oldid=1023950377`, 2021. [Online; accessed 18-June-2021].

[37] S. Xu, S. Venugopalan, and M. Sundararajan. Attribution in scale and space, 2020.

[38] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar. On the (in)fidelity and sensitivity of explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[39] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks, 2013.

[40] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down neural attention by excitation backprop, 2016.