In-Depth Hyperparameter Selection For Layer-Wise Relevance Propagation

vorgelegt von
RODRIGO BERMÚDEZ SCHETTINO
Matrikel-Nr.: 344483

Von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Master of Science
– M.Sc. –

Erster Prüfer:	Prof. Dr. Klaus-Robert Müller
Zweiter Prüfer:	Prof. Dr.-Ing. Thomas Wiegand
Betreuer:	Dr. Grégoire Montavon

Berlin 2022

Erklärung der Urheberschaft

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 5. August 2022

.....
Unterschrift

Abstract

Our expectations of Explainable AI have grown together with its popularity. So far, the interpretability technique of Layer-Wise Relevance Propagation (LRP) has been adopted with mostly qualitative evaluation of its rules. Therefore, a quantitative and qualitative evaluation of LRP rules is conducted to determine which hyperparameters provide the best scoring heatmaps according to the Pixel-Flipping and Area Under the Curve evaluation framework. It can be concluded from the experiment results that the choice of evaluation metrics and visualization of heatmaps has a significant impact on explanations. Additionally, due to the inherent subjectivity of visual explanations the requirements should be defined on a case-by-case basis.

Zusammenfassung

Unsere Erwartungen an Erklärbare KI sind zusammen mit ihrer Popularität gestiegen. Bisher wurden die Regeln der Interpretierbarkeitsmethode Layer-Wise Relevance Propagation (LRP) überwiegend qualitativ bewertet. Im Rahmen dieser Arbeit wird eine quantitative und qualitative Untersuchung der LRP-Regeln durchgeführt, um die Hyperparameter herauszufinden, die die am besten bewerteten Erklärungen liefern. Aus den Ergebnissen lässt sich schlussfolgern, dass die Wahl der Bewertungsmetriken und die Einstellungen der Visualisierung von Heatmaps einen erheblichen Einfluss auf die Erklärungen haben. Zudem liegen visuelle Erklärungen im Auge des Betrachters, was auf die eigene Komplexität der Aufgabe hinweist. Die Anforderungen an Erklärungen sollten im Einzelfall definiert werden.

Acknowledgements

This thesis would not have been possible without the support of many people. First, I would like to thank my advisor, Grégoire Montavon, whose insights helped shape this work and provided guidance to navigate the exciting field of Explainable AI. I truly enjoyed working on this topic and advancing the current state-of-the-art algorithms.

I would like to thank Christopher Anders, Sebastian Lapuschkin, Anna Hedström, and Adrian Hill for our fruitful interactions which contributed to understand certain algorithms better and design the evaluation framework for explanations proposed in this thesis. I would also like to thank my office mate, Niklas Schmitz, for his insights throughout my thesis which especially helped getting up to speed. I thank the ML group members at TU Berlin for our inspiring exchanges.

I am very grateful to Scantist, Prof. Thambipillai Srikanthan, and Prof. Yang Liu for the growth opportunities provided. Prof. Sri advised me from his vast experience, which always provided food for thought, especially in challenging times.

I thank my former supervisor at CERN, Ulrich Schwickerath, for his continuous support and mentoring and whose drive has been inspiring.

On the personal side, I would like to express my gratitude to my family, friends, and Tu-Lien for their unconditional support throughout and to Sebastian Pokutta for his advice during my thesis.

Contents

1. Introduction

Artificial Intelligence (AI) has gained momentum in the past decade and its usage has been increasingly expanded to real-world applications. The first AI algorithms were traditionally accepted as *black boxes*. With time, the desiderata of these systems grew together with their complexity. This motivated looking inside these black boxes to verify that the decisions were well-founded. Also, because the Machine Learning (ML) models became intrinsically more complex, additional, more elaborate training was needed; Explainable AI (XAI) allowed to conscientiously expand these systems by verifying the results and correcting existing errors.

XAI not only allows to demystify these systems which have been mistrusted, but also to conscientiously extend and incorporate them into highly-regulated areas like medicine or construction. Scrutinizing AI has allowed to shine light on ethical topics—e.g., fairness and bias in these systems. These issues originate from the sheer datasets on which these algorithms are trained. After decades of development on these foundations, bias has now been intertwined with these architectures.

XAI has also allowed to uncover Clever Hans predictions (Lapuschkin et al., 2017, 2019; Bykov et al., 2022; Anders et al., 2022; Kauffmann et al., 2020; Samek and Müller, 2019) , which often give the impression of accuracy in these systems by correlating non-relevant features. Furthermore, this also allows to improve datasets used for training and evaluation. XAI helps take these systems accountable and gain a broader acceptance by explaining how they work. Acceptance is particularly important for their further development. However, a discussion about bias, ethics, and compliance is out-of-scope for this thesis.

Neural Networks (NNs) are especially challenging to explain due to their tendency to have deep architectures (see Deep Neural Networks (DNNs)). The LRP algorithm is one proposal to explain NNs results and decompose NN decisions by tracing the output of the network back to the input to identify the input features—pixels, in the case of images—responsible for the network result. This propagation-based explanation framework (Montavon et al., 2019) is not limited to images but in this work we will focus on image classification. The explanations for such systems are given as heatmaps, also called saliency or relevance maps (Tjoa and Guan, 2021).

The main goal of this thesis is to conduct an in-depth exploration of the hyperparameter space of selected LRP rules for DNN using a systematic approach. A by-product is the proposal to standardize the techniques for this specific purpose. In further chapters the challenges in doing so are presented, together with existing options and their shortcomings.

So far, there has been predominantly qualitative evidence, however, we need a quantitative comparison of LRP rules and hyperparameters. This motivates the need of an objective comparison of existing LRP heuristics and a systematic approach to benchmarking visual

explanations.

Outline

The further chapters of this thesis are structured as follows:

- **Chapter 2 (Explainability Background)**: We review taxonomies of explanations with a focus on LRP. Besides, we will discuss the most commonly used LRP rules and where are they applicable. Most commonly used rules with hyperparameters will be presented together with their corresponding heuristics.
- **Chapter 3 (Benchmarking Visual Explanations)**: In the methodology chapter, we present LRP-PF-AUC, the benchmarking framework for generating and quantitatively evaluating heatmaps. It offers a systematic approach to investigate heuristics for LRP explanations; the approach will be validated on sample images.
- **Chapter 4 (Results)**: In the evaluation chapter, we evaluate multiple LRP hyperparameters using the benchmarking framework Pixel-Flipping (PF) and Area Under the Curve (AUC). Next, we look at the structure of experiments and interpret the results. Afterwards, we also look at limitations faced in this thesis and how they could be overcome in future work. Finally, recommendations for LRP rules based on quantitative results are presented.

Main Contributions

The core of this thesis includes the approach on exploring the extensive hyperparameter space at hand and how it is reduced to a feasible and yet meaningful space. The expected result of this thesis are recommendations on the optimal choice of parameters for a given architecture based on a comprehensive evaluation of the alternatives. In other words, the core is to develop a systematic approach to convert this problem into a feasible one.

Concisely, this work provides two key contributions:

- Implementation of framework for streamlined generation and evaluation of heatmaps qualitatively and quantitatively using LRP, PF, and AUC.
- Extensive hyperparameter search based on maximizing PF scores under different constraints.

2. Explainability Background

The main focus of this chapter lies on DNN for image classification to review existing explainability methods and how to evaluate them. For added context, we start by providing an overview of explainability approaches depending on their usage to make a case for the inherent difficulty of producing accurate explanations. In Section 2.1 primarily focuses on LRP; we will present widely used LRP rules together with commonly used heuristics to set their hyperparameters and will show why their choice is not straightforward. Then, proposed techniques to evaluate them objectively will be presented in Section 2.2, together with their shortcomings in Section 2.3.

First, to rate an explanation the requirements for a *good* explanation need to be clearly defined. The desiderata of an explanation consists of fidelity, understandability, sufficiency, low construction overhead, and efficiency (Swartout and Moore, 1993). In other terms, the explanation should accurately represent how the system works, it should be intelligible, informative, fairly straightforward to integrate, and computationally feasible, respectively.

A taxonomy helps choose the right interpretability approach for the respective use case. Moreover, categorization helps understand the limitations of each type. Therefore, a brief overview of interpretability approaches will be given to situate our use case and provide context. Interpretability is especially important in Medical XAI (Tjoa and Guan, 2021), where the requirements for such systems are much higher due to the risks of each potentially wrong decision; after all, to trust these methods, they need to be held accountable (Swartout and Moore, 1993). The categories are not exclusive—i.e., one algorithm can be categorized multiple times according to its properties.

A proposal to categorize interpretability methods is according to either how they are perceived by the user or their mathematical structure (Tjoa and Guan, 2021). Depending on how the explanation is presented to the user, the perceptive interpretations can be subdivided into saliency, signal or verbal. Saliency methods result in an explanation which reflects the contributions of the input. Precisely, LRP (Bach et al., 2015) is considered a saliency method within the perceptive interpretability category. LRP *decomposes* the DNN decision layer-by-layer to produce the explanation.

LRP falls into the category of saliency methods by producing explanations as heatmaps—also called saliency maps—(Tjoa and Guan, 2021), which assign positive or negative values to the contributions in favor or against the network result, respectively. Within the saliency category, LRP corresponds to the decomposition methods because to explain a given NN result, it traces the attributions back through the network to the input to assign the respective relevances, decomposing in its way the original result (Tjoa and Guan, 2021). An alternative LRP is Rate-Distortion Explanation (Macdonald et al., 2020).

LRP is classified as an attribution method within the saliency category because, similarly as in the taxonomy above from Tjoa and Guan (2021), the result is traced back to the original input, decomposing the attribution using a specific set of rules; LRP attributions are called *relevances*. It can also be classified as a structure-based explanation method (Samek, 2019). Another example of a global explanation technique would be Activation Maximization (AM) (Erhan et al., 2009).

2.1. Layer-Wise Relevance Propagation

As previously mentioned, LRP helps generate interpretations of the output of DNNs, these explanations are highly customizable. We will first define LRP, then, a subset of widely used rules in Subsection 2.1.1.

The focus of this section is the explainability algorithm, rather than the choice of NN, thus, the NN will be abstracted by a function $f : \mathbb{R}^d \to \mathbb{R}$. The input for this function f will be $\boldsymbol{x} = (x_1, \ldots, x_d)$. The prediction of the DNN will be $f(\boldsymbol{x})$—further notation included in Table 2.1. R refers to the relevance. z_{jk} is the relevance contribution of j to neuron k. $f(x)$ is the prediction. Heatmaps are also called attribution maps or relevance mappings (Macdonald et al., 2020).

A non-extensive overview of these rules will be provided, focusing primarily on the ones we will investigate in-depth. An extensive investigation of all rules is beyond the scope of this thesis due to the computational resources this would require. In Chapter 3, we will explain the methodology used and present our approach to reduce the hyperparameter space for feasible computation.

Mainly, we will introduce the rules investigated, effectively starting with the general rule (Montavon et al., 2019). A requirement for propagating relevance in LRP rules is the conservation principle (Bach et al., 2015), which requires the same magnitude of relevance to be distributed across the network neurons to avoid losing information when tracing back the relevance from the network decision back to the input. The denominator in Equation 2.2 enforces the conservation property; Equation 2.3 is the general case. Relevance can also be written as the product of activations times factors, as in Equation 2.1.

$$\begin{aligned} R_j &= a_j \cdot c_j \\ R_k &= a_k \cdot c_k \end{aligned} \tag{2.1}$$

$$\text{where } c_j = \sum_k \left(w_{jk} + \gamma w_{jk}^+ \right) \frac{\max\left(0, \sum_{0,j} a_j w_{jk}\right)}{\sum_{0,j} a_j \left(w_{jk} + \gamma w_{jk}^+ \right)} c_k$$

Equation 2.1: Alternative Definition of Relevance.

$$\sum_j R_j = \sum_k R_k$$
$$\sum_i R_i = f(\boldsymbol{x}) \tag{2.2}$$

Equation 2.2: Layer-Wise and Global Conservation Property.

Notation	Description	Notation	Description
$\boldsymbol{x}$	Input	z_{jk}	Rel. contr. of neuron j to k
$f(\boldsymbol{x})$	Prediction	$\boldsymbol{l}, \boldsymbol{h}$	bounds used by first layer in z^B
R	Relevance	$\hat{\nabla}$	gradient of $f(x)$ with detached terms
w_{jk}	Weights and biases	f_1^+, f_1^-	forward passes on copies of first layer
w_{0k}	Neuron bias		whose parameters were processed
a_k	Neuron activations		by $\max(0,\cdot)$ and $\min(0,\cdot)$
$a_0 = 1$	By definition	$\theta \mapsto \rho(\theta)$	Hyperparameters
$(a_j)_j$	Lower-layer activations		

Table 2.1: LRP Notation.

2.1.1. LRP for Deep Rectifier Networks

In this subsection, we will summarize the work in Montavon et al. (2019). The definition of LRP rules is model-dependent; these rules are also different depending on the layer where they are implemented (Kauffmann et al., 2019). Examples of different layer types are linear, pooling, pixel, and fully-connected layers. We will focus on feed-forward NNs because we understand them relatively well.

First, we will define an abstraction for DRN neurons in Equation 2.4. Although there are multiple LRP rules, we will focus on a subset of rules, which will allow us to conduct an in-depth evaluation given the time and resource constraints of this thesis. The rules we will use are defined in Table 2.2. LRP_ε extends LRP_0 by the additional term ε to avoid division by zero and reduce noise inversely proportional to the magnitude of ε, thus, absorbing low-relevance scores. In LRP_γ, γ favors positive contributions over negative ones and has a *smoothing* property (Montavon et al., 2019). The z^B rule(Montavon et al., 2017)—pronounced *box rule*—is designed for the first layer, also called the pixel layer. The terms l_i, h_i refer to the lowest and highest pixel values of x_i, where x_i is the input image to the NN.

2.1.2. Recommendations for LRP Rules

There are different terms to refer to the position of layers in the network. The *pixel* layer is referred to as *upper* layer or *first* layer. The *middle* layer is self-explanatory. The term *lower* layer refers to layers next to output, also called *last* layers; it is important to make this distinction

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{2.3}$$

Equation 2.3: General LRP Rule.

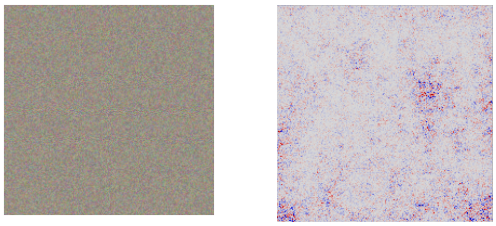
$$a_k = \max\left(0, \sum_{0,j} a_j w_{jk}\right) \tag{2.4}$$

Equation 2.4: Neurons in Deep Rectifier Network (DRN).

because LRP backpropagates the relevance, which can lead to confusion when referring to the position of layers because the order is reversed as opposed to producing an output from the input. In Table 2.4, n refers to all subsequent layers in the model.

Before we proceed to the rules, we need to define the vocabulary to refer to different combinations of LRP rules. A *uniform* LRP rule refers to the application of only one rule to the whole network. In contrast, a *composite* in the context of LRP refers to a combination of rules; these can be defined via layer index or layer type (see Table 2.4, Table 2.3, Table 2.4).

For the sake of completion, Figure 2.1 shows the application of LRP to a random input image. The basic explanation technique Gradient × Input—shown in Figure 2.2b—is equivalent to $\text{LRP}_{\gamma=0}$, $\text{LRP}_{\varepsilon=0}$, and LRP_0 (Shrikumar et al., 2016). The castle image (Montavon, 2021) depicted in Figure 2.2a is used to compare qualitatively the equivalence Gradient × Input in Figure 2.2b with Uniform LRP_0 in Figure 2.2c (Samek et al., 2019). However, applying the z^B rule for the pixel layer modifies the equality above and reduces the noise of the explanation significantly, as seen in Figure 2.2d. Table 2.5 is a summary of (Kohlbrenner et al., 2020; Montavon et al., 2019; Andéol et al., 2021).



(a) Random Input **(b)** Uniform LRP_0

Figure 2.1: LRP on Random Image.

2.1.3. Advantages and Limitations of LRP

The advantages of LRP are that it provides *global*, *continuous*, *image-specific* explanations with *positive and negative evidence* and it allows for *aggregation over regions or datasets* and its *mathematical relationship between network output and explanation* (Samek et al., 2017).

LRP offers a trade-off between computation and robustness; methods relying on gradients are often not robust. Perturbation/sampling methods are robust but slow, whereas LRP is fast and robust. The challenge of LRP is the implementation overhead of rules, which depends on

LRP Rule	Definition	$\rho(w_{jk})$	ε
Generic $\text{LRP}_{0/\varepsilon/\gamma}$	$R_j = \sum_k \frac{a_j \cdot \rho(w_{jk})}{\varepsilon + \sum_{0,j} a_j \cdot \rho(w_{jk})} R_k$		
LRP_0	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$	w_{jk}	0
LRP_ε	$R_j = \sum_k \frac{a_j w_{jk}}{\varepsilon + \sum_{0,j} a_j w_{jk}} R_k$	w_{jk}	
LRP_γ	$R_j = \sum_k \frac{a_j \cdot (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j \cdot (w_{jk} + \gamma w_{jk}^+)} R_k$	$w_{jk} + \gamma w_{jk}^+$	0
z^B	$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{wj}^+ - h_j w_{jk}^-}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_j w_{ij}^-} R_j$		

Table 2.2: Definition of LRP Rules for DRN.

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
2-10	LRP_γ	$\gamma = 0.5$
11-17	LRP_γ	$\gamma = 0.25$
18-24	LRP_γ	$\gamma = 0.1$
25-31	LRP_γ	$\gamma = 0$

Table 2.3: Definition of $\text{LRP}_{\text{Decreasing}-\gamma}$ (Eberle et al., 2022).

the architecture. LRP is an advanced explainability method, hence, when a simple explanation would suffice, other methods might be better suited for the task—see Figure 2.2—but LRP is able to provide a performance boost for more complex tasks.

2.1.4. Practical Considerations

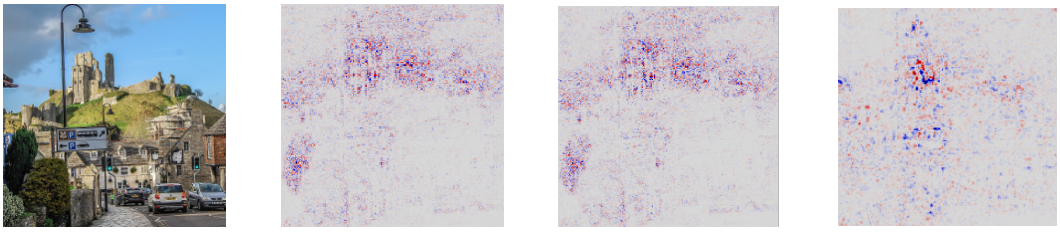
Neural Network Canonization

An implicit requirement for generating LRP explanations is that the NN should be in canonical form; to the best of our knowledge, there is no formal definition of canonical form. This, however, has been briefly mentioned before in (Binder, 2020; Letzgus et al., 2021). In this section, a literature review on canonization of DNNs for LRP was conducted and a summary of implicit assumptions is given. For a NN to be in canonical form there has to be an alternation of linear/convolution, Rectified Linear Unit (ReLU), and pooling layers (Letzgus et al., 2021).

It is not always possible to convert a NN into a canonical form. The VGG model without

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
1-16	LRP_γ	$\gamma = 0.25$
17-30	LRP_ε	$\varepsilon = 0.25$
31-n	LRP_0	

Table 2.4: Definition of $\text{LRP}_{\text{Tutorial}}$.



(a) Input **(b)** Gradient × Input **(c)** Uniform LRP_0 **(d)** z^B and LRP_0

Figure 2.2: Comparing Basic Explanations.

batch normalization layers is already in canonical form. To bring a model into its canonical form, we can convert layers into an equivalent form without altering their output—e.g., the $\text{LRP}_{\text{Tutorial}}$ implementation (Montavon, 2021) converts dense layers to convolutional during canonization. Furthermore, canonization is implemented in (Anders et al., 2021).

Numerical Stability

LRP is robust against gradient shattering (Montavon et al., 2018). Heuristics are required when implementing LRP to ensure numerical stability (Montavon et al., 2019). We will provide a brief overview of common pitfalls when implementing LRP and we will also show how they can be solved and the steps we took to ratify our LRP implementation used in Chapter 3 and Chapter 4.

We need to ensure numerical stability when implementing LRP (Montavon et al., 2017). The ε-stabilized denominator was first proposed by Bach et al. (2015); a concise overview of the possible *stabilizer* terms will be provided. We encountered that only setting $\varepsilon > 0$ in the denominator does not result in a numerically stable implementation in several cases, as shown in Figures 2.3, 2.4, 2.5.

For brevity, we will add annotations as exponents to the composites in this section—e.g., $\text{LRP}^{\text{vanilla}}_{\text{Tutorial}}$ for an implementation without heuristics, hence *vanilla*. In Figure 2.3 we observe how numerical instability becomes evident in the heatmap if no heuristic is used to implement the generic $\text{LRP}_{0/\varepsilon/\gamma}$ rules, which are all part of the $\text{LRP}_{\text{Tutorial}}$ composite. However, we encounter that this is image-dependent, as the same implementation (without heuristics) calculates a valid heatmap in Figure 2.4 with a different input. Numerical instability

Rule	Layer	Hyperparameters
LRP_0	Upper	
LRP_ε	Middle	$\varepsilon < 1$, $\varepsilon = 0.25$
LRP_γ	Lower	$\gamma < 1$, $\gamma = 0.25$
z^B	First (pixels)	

Table 2.5: Overview of LRP Recommendations (Montavon et al., 2019).

can also occur when applying the zennit implementation of LRP; ongoing discussion under https://github.com/chr5tphr/zennit/issues/148.

The numerical instability is manifested by patches in the heatmap where the relevance scores concentrate by several orders of magnitude. This happens when the conservation property of LRP is violated but it can be prevented by using a heuristic in the implementation of the rules $\text{LRP}_{0/\varepsilon/\gamma}$. While certain heuristics are numerically stable in a specific setup, it is important to integrate automated verification into the implementation.

In our initial investigation into how to automate this verification, we calculated the standard deviation of the ImageNet dataset, explored different parameter values for hyperparameter γ to verify if the choice of hyperparameter had an influence on the occurrence of numerical instability. The patches in the heatmap are reproducible with different composites—e.g., uniform $\text{LRP}_\gamma^{\text{vanilla}}$ or composite $\text{LRP}_{\text{Tutorial}}^{\text{vanilla}}$ with variable γ values. These patches might be reproduced for ε, analogously. Figure 2.3c shows the influence the plotting settings have on the display of patches.

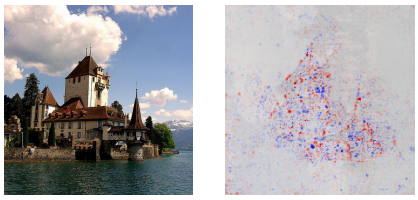
The grid search in Figure 2.5 showcases different heatmaps from the same LRP implementation without heuristics in the denominator of $\text{LRP}_{0/\varepsilon/\gamma}^{\text{vanilla}}$ in Equation 2.5.

(a) Input **(b)** Heatmap **(c)** Heatmap

Figure 2.3: $\text{LRP}_{\text{Tutorial}}^{\text{vanilla}}$ for castle image.

2.1.5. Heuristics in Denominator of Relevance Calculation

To improve numerical stability in LRP, several heuristics have been proposed; the implementation of $\text{LRP}_{\text{Tutorial}}$ (Montavon, 2021) uses quadratic mean, the zennit framework (Anders et al., 2021) sets all zero elements in denominator to an ε value. The heuristic used in the definition of the LRP_ε rule in the composite $\text{LRP}_{\text{Tutorial}}$ to scale z proportionally to ε is to replace



(a) Input **(b)** Heatmap

Figure 2.4: $\text{LRP}^{\text{vanilla}}_{\text{Tutorial}}$ for *castle2* image.

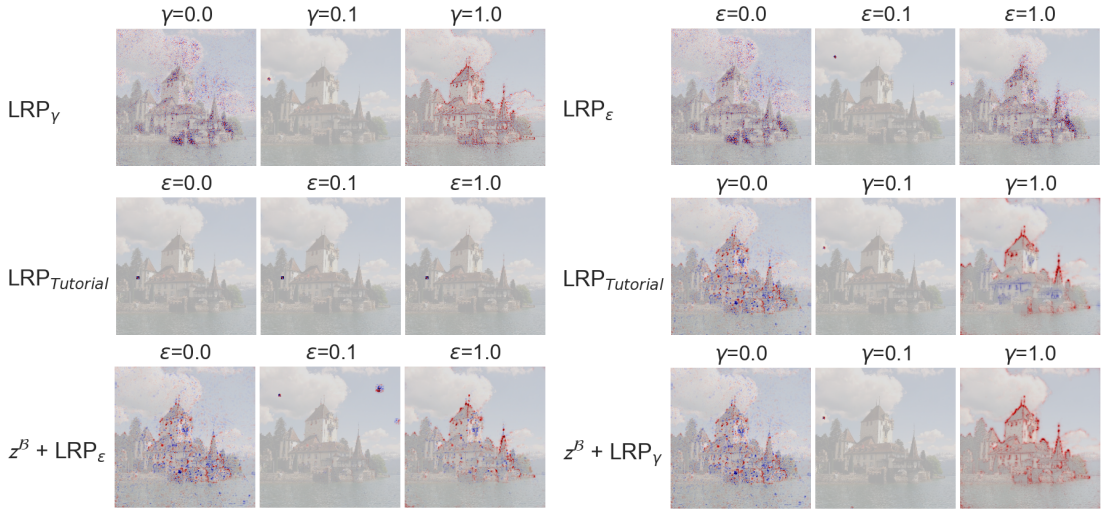


Figure 2.5: Numerical instability in grid search for castle2 image without heuristics.

it by $\varepsilon \cdot \sqrt{z^2.\text{mean}()}$. A non-exhaustive list of heuristics in pseudocode for the denominator in Equation 2.5:

1. **Vanilla.** No heuristics:

   ```
   epsilon + dividend
   ```

2. **Zennit.** Add epsilon to the absolute value of the dividend conserving the sign:

   ```
   dividend + ((dividend == 0.).to(dividend)+ dividend.sign())* epsilon
   ```

3. **$\text{LRP}_{\text{Tutorial}}$.** Scale epsilon according to dividend's magnitude using quadratic mean:

   ```
   dividend + epsilon * (dividend**2).mean()**.5 + 1e-9
   ```

2.2. Pixel-Flipping and Region Perturbation

There have been attempts to define a universal metric for heatmaps—e.g., file size metric (Samek et al., 2021), rate-distortion (Macdonald et al., 2020)—but the search continues, in

$$R_j = \sum_k \frac{a_j w_{jk}}{\varepsilon + \sum_{0,j} a_j w_{jk}} R_k \tag{2.5}$$

Equation 2.5: Denominator where numerical stability should be enforced.

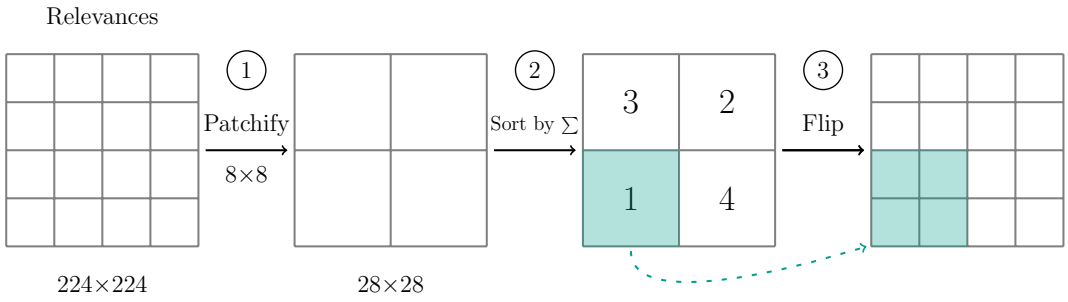


Figure 2.6: Pixel-Flipping Algorithm

part, due to the intrinsic complexity of the problem. Depending on the audience, the explanation requirements vary. Rating NN explanations using perturbation has become a popular approach (Fong and Vedaldi, 2017; Agarwal and Nguyen, 2021; Macdonald et al., 2020). Pixel-Flipping (Bach et al., 2015; Samek et al., 2017), also called *relevance ordering based test*—is an iterative metric for faithfulness/sufficiency of heatmaps, which measures the rate of decay of the classification score of a NN when the explanation is progressively *destroyed* or *flipped*.

The algorithm has developed multiple variants, pixels can be either *flipped*—i.e., replaced—by a constant or random value or can be replaced using generative models—e.g., inpainting—, as an improvement to avoid deviating too strongly from the input image. Initial versions of PF flipped pixels to gray color—i.e., the intermediate value in the range of values allowed for the pixels. Another parameter of the algorithm is how to *sort* the relevance scores of the heatmap, in other words, in which order should the pixels be destroyed.

The available *sorts* for PF are *Most Relevant First (MoRF)*, *Least Relevant First (LRF)*, and *Random*, and neutrally predicted first, where values with absolute value closest to zero are flipped first (Bach et al., 2015). The most widely used modes are *inpainting* and *random*. The PF algorithm is described in Figure 2.6 for an image of 224×224 dimensions and patch size of 8×8. The term *patchify* refers to the process of dividing the relevance scores into a grid of patches of a certain size.

There is yet to be a consensus on the standard name of PF, which is why we will give an overview of the different terms used for this algorithm with slight degrees of variation. First, the notation we will adopt for specifying the PF mode and sort is PF^{Mode}_{Sort}. E.g., for PF mode *inpainting* and sort *MoRF* we will write $\text{PF}^{Inpainting}_{\text{MoRF}}$.

PF is also called Region Perturbation (RP), originally for perturbation sizes larger than a single pixel, although in literature the term PF is frequently used interchangeably for different perturbation sizes. PF_{MoRF} is also called pruning curve, PF_{LRF} activation curve; PF_{LRF} focuses

predominantly on negative features, contrary to PF_{MoRF}, which focuses on positive ones and controls how simple an explanation is. For a given number of pixels, decrease of function is a trade-off between faithfulness and interpretability.

In case there are relevances of the same magnitude in the array of relevance scores, they need to be replaced individually to accurately measure perturbations. Sampling range for flipped values must include also negative values, otherwise, positive relevances are favored, which results in an artificial image. The preferred method for flipping values is by inpainting instead of by random sampling because inpainting is more natural and avoids creating artifacts (Samek et al., 2021). In the last perturbation step, we choose to flip all pixels to gray color to ensure that the last classification score is closest to zero.

Flipping individual patches creates an artificial image, patches in RP resemble natural occlusion. One of the decisions is whether the patches should be overlapping or not. Flipping patches requires heuristics to be implemented and cover the edge cases, e.g., patch size which is not equally divisible by the dimensions of the input.

Code Listing 2.1 shows the pseudocode from Samek et al. (2019), Code Listing 2.2 shows the implementation with masks; `flip_value` refers to the value to replace pixels in the original image.

Code Listing 2.1: Pseudocode PF (Samek et al., 2019).

```
Sort pixels / patches by relevance
Iterate
  destroy pixel / patch
  evaluate f(x)
Measure decrease of f(x)
```

Code Listing 2.2: Mask PF.

```
s = sort(R)
x[R > s[i]] = flip_value
```

The term *flipping* refers to the replacement of pixels in the original image at the indices corresponding to the relevance scores in all three channels for the case of RGB colors. E.g.: $\text{RGB}(i, j, k) \rightarrow \text{RGB}(a, a, a)$. The value a is the value to replace the pixels with, its calculation depends on the perturbation mode. For perturbation mode random, an option is to sample from the uniform distribution. An alternative is to sample from the existing minimum and maximum bounds of the input.

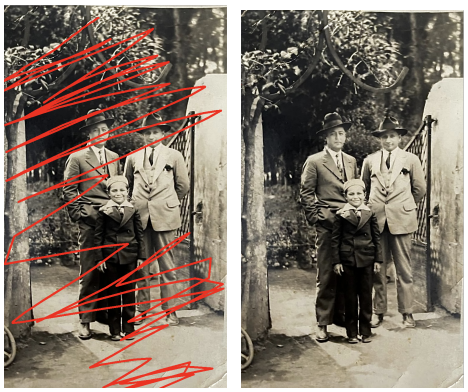
To define RP for perturbation sizes larger than one, there is a detail to consider. Defining perturbation size as k, a heatmap is considered to be a matrix of relevance scores $m \times n$. To divide a heatmap into a grid of patches, we proceed to check if it can be evenly divided by the dimensions of the heatmap. Then, each patch of size k receives a score which is the sum of individual scores in that region. Next, sort patches according to pixel-flipping objective (e.g., MoRF, as shown in Figure 2.6. Finally, progressively remove patches and measure delta in NN results.

It is more efficient to remove multiple steps at once in a logarithmic manner than a linear scale to skip uninteresting parts and optimize for interesting ones in the perturbation process. For sort MoRF, the most interesting part of the flipping curve is the beginning, where we expect the curve to fall the fastest. Quite the contrary for the sort LRF, where we expect the flipping curve to remain constant for most of the perturbation process and fall drastically at the end, where positive relevances of largest magnitude are perturbed. Another argument in favor of flipping multiple patches per step is the amount of computational resources needed to evaluate the classification function at every step otherwise.

To motivate the perturbation curve to remove as much from the original image as possible for MoRF sorting, previously perturbed regions should be flipped repeatedly in subsequent perturbation steps.

2.2.1. Inpainting

The perturbation mode *inpainting* comes from the term with the same name, originally referring to reconstructing damaged or lost parts of images. It aims to provide an accurate filling for the regions to inpaint. It uses image information to fill the regions (Telea, 2004). Figure 2.7 shows an example of a damaged picture in Figure 2.7a, damaged regions are displayed in red, and the inpainting result in Figure 2.7b. Figure 2.8 showcases the effects of inpainting on our reference image, castle. Inpainting contains desirable properties for a perturbation technique because regions are flipped in a more natural way than with random sampling.



(a) Damaged **(b)** Inpainted

Figure 2.7: Damaged picture inpainted with Telea (2004).

2.3. Comparing Visual Explanations

Our aim is to compare multiple visual explanations with each other. We reviewed PF and RP in Section 2.2, which produce a perturbation curve for each explanation based on the delta of classification scores after perturbing the input based on the relevance scores obtained with an XAI algorithm such as LRP. Naturally, we can average between multiple perturbation curves but to compare between explanations, ideally, we would like to grade each one by score.

(a) Quarter **(b)** Patch

Figure 2.8: Inpainted castle image using algorithm by Telea (2004).

In this section we will talk about previously proposed benchmarks and the state-of-the-art. Area Over the Perturbation Curve (AOPC), also called Area Over the Curve (AOC), (Samek et al., 2017) was coined as a metric to compare different heatmaps with MoRF sort. However, this quantity is not inherently bounded and requires further heuristics in practice. Area Under the Perturbation Curve (AUPC) or AUC provides an alternative which is naturally bounded by zero. Related terms are Area Under the Activation Curve (AUAC) and Area Under the Flipping Curve (AUFC).

PF_{MoRF} is sometimes also referred to as *pruning* and the resulting perturbation curve as *pruning curve*. Analogously, PF_{LRF} is also called *activation*. Less AUAC is better for pruning curves, and higher AUAC is better for activation curve (Ali et al., 2022). We will refer to this metric as AUC because the term is applicable to different sorts of PF.

3. Benchmarking Visual Explanations

In this chapter we will introduce our evaluation framework for benchmarking visual explanations. It consists of existing algorithms and metrics with custom modifications to improve their performance. Also, we will motivate our reasoning behind these choices and give a glimpse into the experiments conducted in Chapter 4.

The model we used to test the proposed framework is VGG-16 (Simonyan and Zisserman, 2015), as it is arguably one of the most widely used architectures, but the framework is also applicable to other models. Moreover, the effect of LRP rules on VGG-16 has been previously studied (Montavon et al., 2019). Figure 3.1 of VGG was illustrated using PlotNeuralNet (Iqbal, 2018); ReLU units are omitted for brevity, similarly as in Simonyan and Zisserman (2015). We use the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 (Russakovsky et al., 2015) dataset for large-scale experiments in Chapter 4 and to validate the evaluation framework in this chapter.

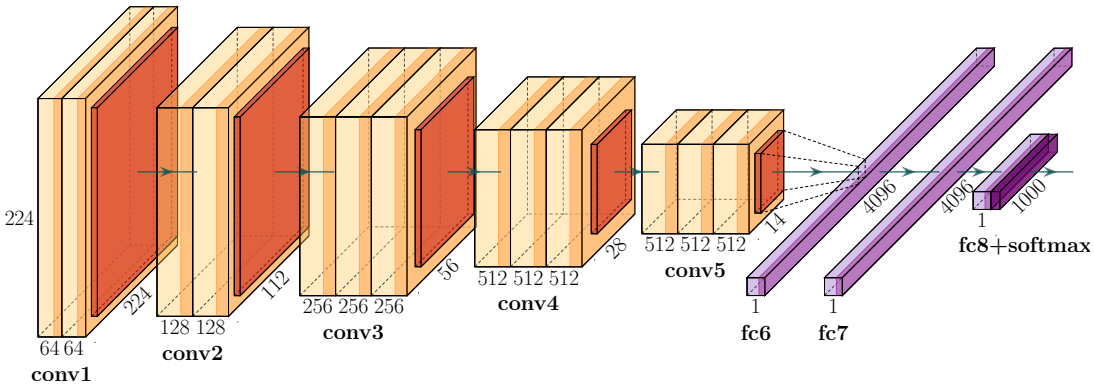


Figure 3.1: VGG-16 Model for ILSVRC dataset, with 1000 neurons in the output layer.

We focus on explanations of Computer Vision (CV) using feed-forward NN with ReLU nonlinearities as LRP rules have been previously tested for this use case. Previous work applies heuristics to generate the LRP explanations; these heuristics will be evaluated with the goal of arriving at optimal hyperparameters to use for future work.

It is possible to explain a different class than the ground truth label using LRP and PF. For the experiments, we opted to explain the ground truth class. To validate the approach, the experiments were executed locally as well as on the cluster, on a larger scale. Locally, experiments ran using shell scripting; to optimize their performance, parallel GNU was used (Tange, 2022).

To verify our LRP implementation (Bermudez Schettino, 2022), we used PF and compared the heatmaps qualitatively with zennit's LRP implementation. We also verified the conservation property defined in Equation 2.2 of relevance between layers and calculated the maximum and minimum relevance scores—as shown in Figure 3.2—and compared them between different heatmaps. We also calculated the standard deviation of a subset of explanations.

Although the LRP algorithm has already been previously implemented (Anders et al., 2021; Lapuschkin et al., 2016; Alber et al., 2018), during the scope of this thesis, LRP, PF, AUC were implemented in Bermudez Schettino (2022), with improvements for the latter two.

There are already implementations of PF, mostly using a linear approach for the perturbation steps. In this chapter we will introduce our proposed improvements and their rationale. It is worth noting that XAI explanations and evaluations need to be in line with the use case. Given the sheer amount of hyperparameters, it is inherently difficult to have one-size-fits-all approach. Even so, often explanations require additional *explanation* and are only valid within a certain context (Hedström et al., 2022).

3.1. Narrowing Down the Hyperparameter Space

In this section, we will explain the approach to explore the hyperparameter space. The naïve approach would be to randomly search all possible hyperparameter combinations. This would be sub-optimal because it would involve exploring the available LRP rules and their hyperparameters. Additionally, given the limited time and computational resources available and the options to explore, which include type of architecture, dataset, choice of LRP composites and their hyperparameter values, and choice of evaluation metrics, we decided to focus on the VGG-16 model, ILSVRC dataset and two LRP composites.

Another consideration is whether assigning LRP rules by layer index provides an advantage over filtering by layer type instead. Assigning rules by layer index allows to treat relevances in different parts of the network differently, as discussed in Montavon et al. (2019). The implicit assumption is that the same rule can have a different effect depending on the position of the layer.

The LRP composites we selected to evaluate are: $\text{LRP}_{\text{Tutorial}}$— Table 2.4—and $\text{LRP}_{\text{Decreasing}-\gamma}$ (Eberle et al., 2022)—Table 2.3—, which is a simpler alternative to the first composite because involves implementing only one LRP rule and only having the hyperparameter value as variable. $\text{LRP}_{\text{Tutorial}}$ represents a more complex composite given the four different LRP rules it leverages. We would like to evaluate whether this added complexity results in better explanations or whether the simplicity of $\text{LRP}_{\text{Decreasing}-\gamma}$ is enough to produce meaningful explanations. The castle image from the original $\text{LRP}_{\text{Tutorial}}$ (Montavon, 2021) was used throughout as a reference image to compare results with other implementations.

For reference, there are variations of the $\text{LRP}_{\text{Decreasing}-\gamma}$ composite. One would be an exponential decay of γ hyperparameter throughout layers, as opposed to halving γ at every block or dividing it by a pre-defined factor.

To discard irrelevant regions for the large-scale experiments, we started by conducting exploratory grid searches, initially on previously proposed ranges—e.g., near zero (Eberle et al.,

2022). Therefore, a grid search was conducted in Section 3.2 to evaluate hyperparameters qualitatively and develop an intuition over the hyperparameters, as to reduce the hyperparameter space to a more relevant subspace.

The hyperparameters are theoretically defined in the positive integer range until infinity, but, empirically, the most interesting range is near zero, where most changes happen. Afterwards, the changes stagnate and become monotonic; further examples in Appendix B. Therefore, a logarithmic scale was used instead of a linear one to discover the area near zero with more detail than the latter.

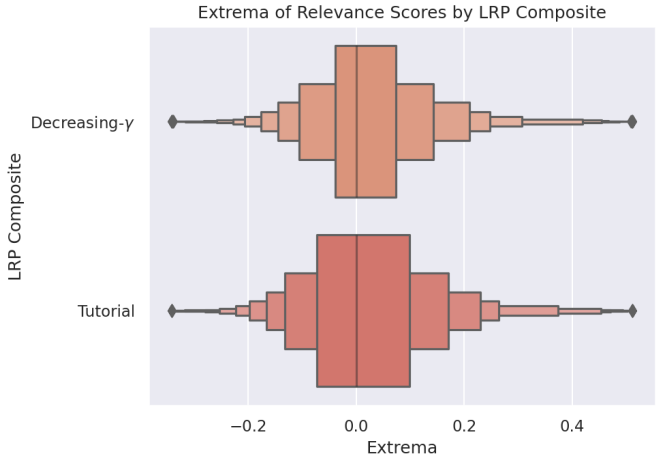


Figure 3.2: Minima and maxima of relevance scores in final experiments.

3.2. Qualitative Evaluation of Heatmaps

An initial grid search was conducted to evaluate the effect of the individual hyperparameters both for uniform and composite applications of LRP rules. This contributed to the choice of hyperparameters tested both quantitatively and qualitatively in Chapter 4.

Eberle et al. (2022) suggests that γ values have the most noticeable impact when closest to zero but not zero, which is why we decided to sample values logarithmically between zero and one. In Figure 3.3, we observe that ε reduces the noise of the attributions as it becomes larger in magnitude by absorbing weaker relevance. This is congruent with the ε definition (Montavon et al., 2019). Effects of the ε parameter are especially present on the lamp post at the top of the castle image for $\varepsilon = 1.0$ in uniform LRP heatmaps. We notice that the differences in between ε values are mostly negligible. It is evident from the heatmaps that z^B significantly improves the quality of the generated explanations. Additional results are included in Appendix B.

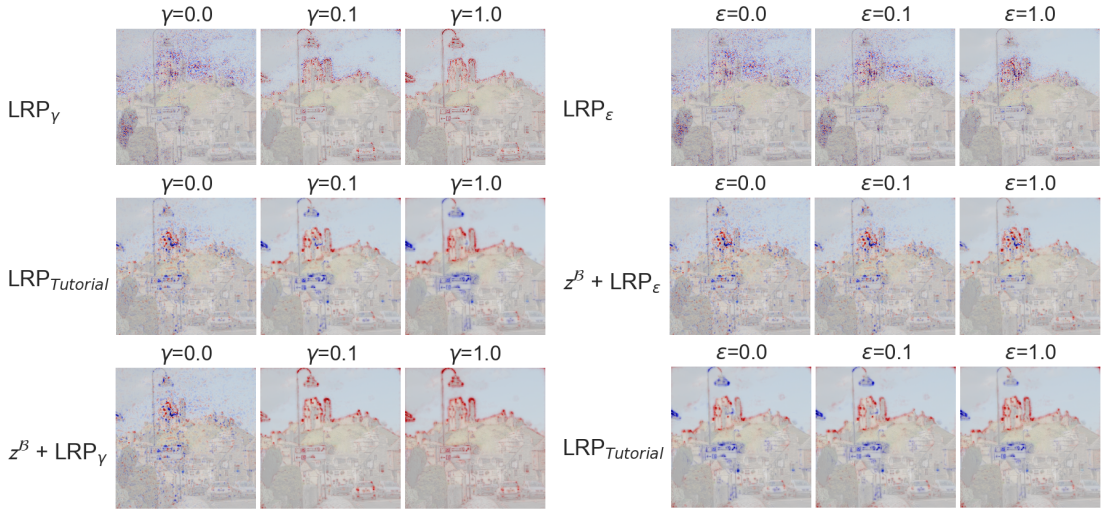


Figure 3.3: Grid search for castle image.

3.3. Quantitative Evaluation of Heatmaps

In this section we will show the original PF algorithm presented in Section 2.2 and motivate the improvements proposed—i.e., multiple flips per step and repeteadly perturbing already perturbed regions in subsequent steps. Perturbation is applied to non-overlapping regions and regions are flipped using the inpainting algorithm by Telea (2004).

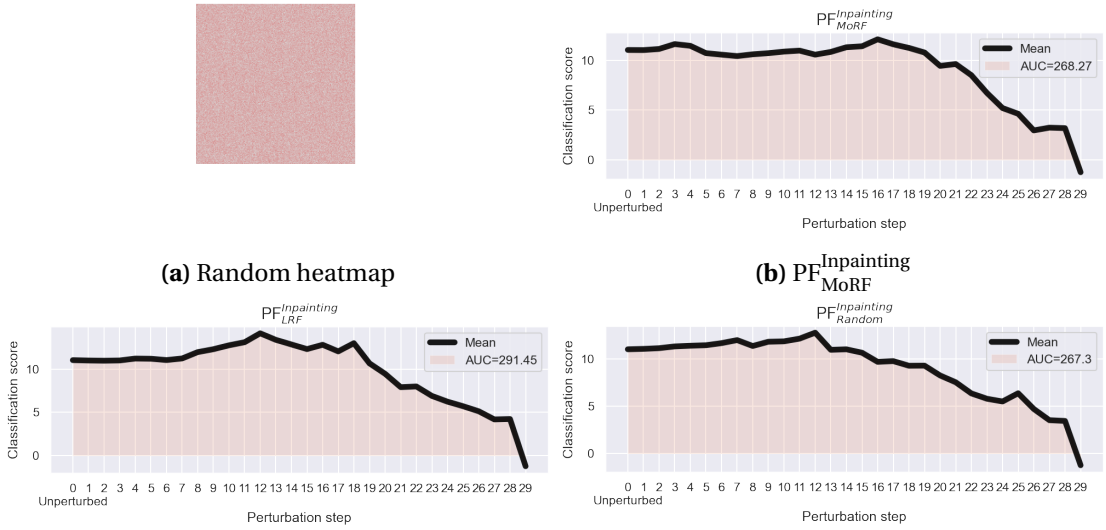
Given multiple LRP-generated heatmaps, we would like to categorize which is the best one by leveraging the LRP-PF-AUC framework. We use PF to compute their corresponding perturbation curves; to compare them, it is desirable to reduce them to a single value to grade the explanations. We have seen in Section 2.3 that AOPC requires further heuristics because it is inherently unbounded. We prefer AUC for its natural boundedness of zero, which translates to integrating under the curve to obtain AUC.

In summary, the proposed quantitative evaluation framework comprises PF for quantifying the quality of heatmaps and reducing the perturbation curve to a AUC score. We provide an example of our evaluation framework applied to the castle image to validate its functionality in Figure 3.5 and also to randomly generated relevance scores in Figure 3.4; values were sampled from a uniform distribution on the following interval: $[0, 1)$.

In Figure 3.6a and 3.6b we can observe that the perturbation curve for PF_{LRF} is in line with its definition. Removing least relevant pixels first should result in largely unchanged classification scores until pixels with stronger relevances are perturbed at the end, where we observe the decline in the curve.

3.3.1. Improving PF Performance

In Figure 3.7a we notice that flipping pixels using randomly sampled values results in an artificially looking image with no resemblance to the original input image. On the other hand,



(a) Random heatmap

(b) $\text{PF}^{\text{Inpainting}}_{\text{MoRF}}$

(c) $\text{PF}^{\text{Inpainting}}_{\text{LRF}}$

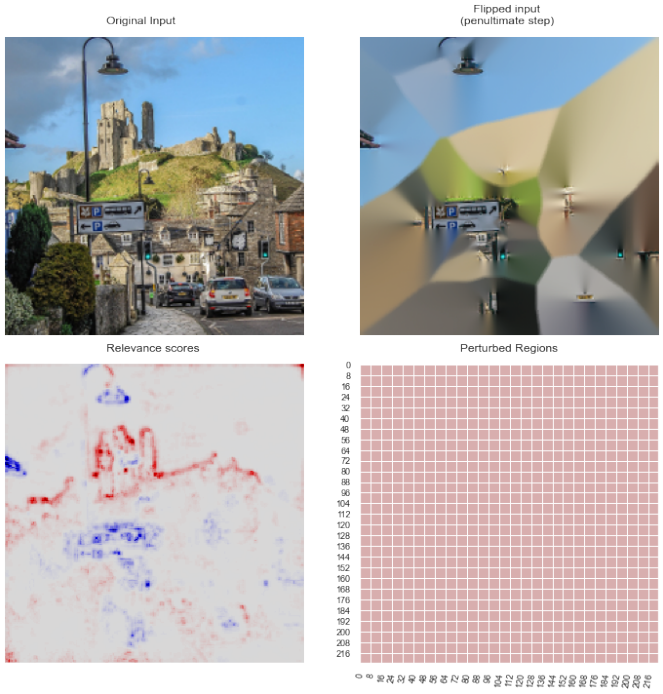
(d) $\text{PF}^{\text{Inpainting}}_{\text{Random}}$

Figure 3.4: PF applied to random relevance scores.

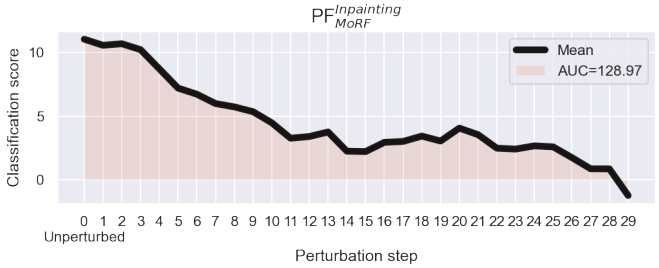
it can be observed in Figure 3.5a that flipping regions using inpainting results in better AUC scores and the image preserves some resemblance to the original image. In Figure 2.8, we show the results of the inpainting algorithm, integrated into PF.

For the sake of completion, we show the perturbation curves of $\text{PF}^{\text{Random}}$ in Figure 3.7. We notice that the perturbation curves of $\text{PF}^{\text{Random}}$ do not match the expectations of MoRF, LRF, and Random, respectively.

The proposed improvements to the state-of-the-art PF algorithm is the number of simultaneous flips per perturbation step, as shown in Figure 3.8. Note that these were calculated for an image of dimensions 224×224. For perturbation size 8×8, the dimensions become 28×28. Therefore, the maximum number of flips for this specific example is 784—the total number of patches of size 8. The initial step is to measure the classification score of the unperturbed image to compare it against the perturbations later on. The last step is a gray image to destroy any resemblance of the initial image, as shown in Figure 3.9, hence the 784 number of flips in the final PF step. The goal of the granular steps is to measure consistently the expected changes in classification scores as the image is progressively perturbed.

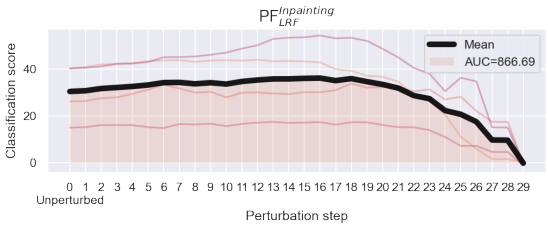


(a) Comparison of LRP-generated heatmap with PF-perturbed input image

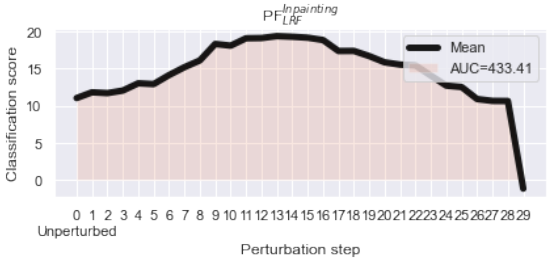


(b) AUC

Figure 3.5: $\text{LRP}_{\text{Tutorial}}$-$\text{PF}^{\text{Inpainting}}_{\text{MoRF}}$-AUC evaluation of castle image.

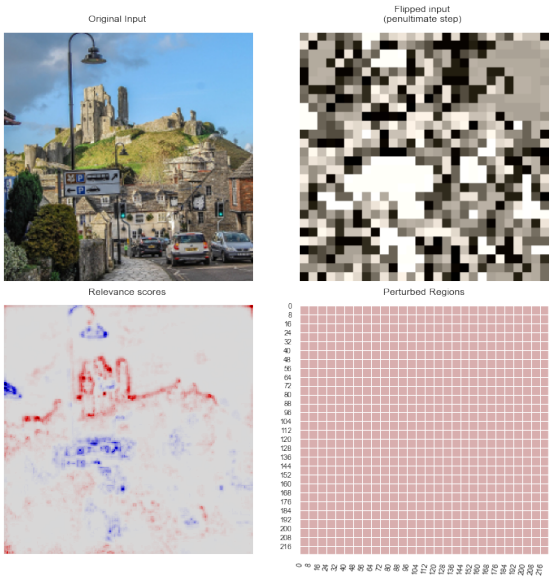


(a) $\text{LRP}_{\text{Decreasing}-\gamma}$-$\text{PF}^{\text{Inpainting}}_{\text{LRF}}$-AUC of batch of 4 *axolotl* images

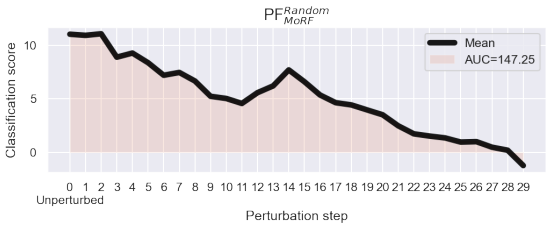


(b) $\text{LRP}_{\text{Tutorial}}$-$\text{PF}^{\text{Inpainting}}_{\text{LRF}}$-AUC for castle image

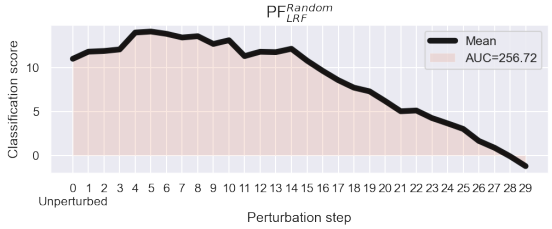
Figure 3.6: $\text{PF}^{\text{Inpainting}}_{\text{LRF}}$ on castle and *axolotl* images.



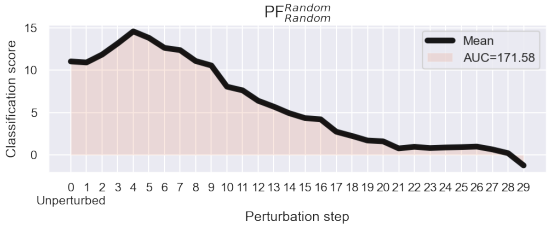
(a) Comparison of LRP-generated heatmap with PF-perturbed input image



(b) PF^{Random}_{MoRF}



(c) PF^{Random}_{LRF}



(d) PF^{Random}_{Random}

Figure 3.7: $LRP_{Tutorial}$-PF^{Random}-AUC evaluation of castle image.

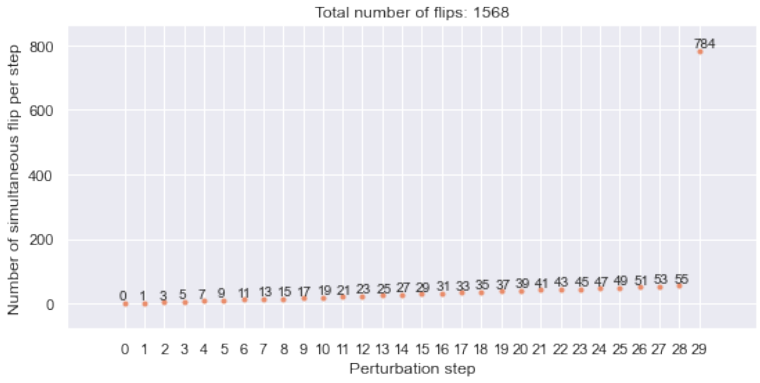


Figure 3.8: PF - Number of flips per step for 224×224 image divided into 8×8 patches.

Figure 3.9: Last Step of PF Algorithm: Gray Image.

4. Results

In this chapter, the results from applying the evaluation framework proposed in Chapter 3 are reported in Section 4.1. LRP composites are evaluated and compared with each other. Then, the optimal heatmaps according to our results are presented in Section 4.2, Figure 4.3. Finally, these findings are compared to existing heuristics and we come to a conclusion about the interpretation of said optima.

The experiments were conducted using the Neural Network model VGG-16 model and the ILSVRC 2012 dataset; details of the dataset used can be found in Appendix A. We used all 50 images available in the dataset from the *axolotl* class in the experiments for simplicity's sake. The hardware used includes NVIDIA A100 40GB PCIe GPU. The experiments were executed on a cluster with nodes with a minimum of 12GB virtual memory and a minimum of 12GB of free memory.

As mentioned earlier, the quantitative evaluation is performed using LRP for explaining NN decisions. Then, PF is used to evaluate the quality of these evaluations. Finally, the perturbation curve of such procedure is reduced to a numerical score, which can be used to compare several explanations, leveraging AUC. The approximate elapsed time for calculations in the experiments is as follows: 4 seconds for the NN forward pass, 4 seconds for the LRP explanation, 5 minutes for the PF process. Inpainting is arguably one of the most computationally expensive operations in the experiments.

4.1. Quantitative Results

To summarize the experiment results we will leverage contour plots, where the hyperparameters tested are located on the x- and y-axes and their corresponding AUC score is displayed on the z-axis by color intensity. The goal of these contour plots is to identify optimal hyperparameters according to our evaluation framework. In the discussion part, we will compare the results achieved to the commonly established heuristics for LRP hyperparameters. We are limited to two variables in the contour plots, due to computational constraints and also due to natural visualization constraints.

For $\mathrm{PF}_{\mathrm{LRF}}$, higher AUC scores are better; for $\mathrm{PF}_{\mathrm{MoRF}}$, lower AUC scores are better. By definition, we expect experiments with $\mathrm{PF}_{\mathrm{random}}$ to result in constant contour plots—i.e., constant AUC score and single color—because the choice of LRP hyperparameters is independent of the PF performance. A constant AUC score means that LRP hyperparameters are not correlated with PF performance.

The hyperparameter values were sampled logarithmically with base 10, with zero addition-

ally. Altogether, the hyperparameter space was defined as $\{0, 10^{-4}, \ldots, 10^{0}\}$; a total of 16 values were used as hyperparameter space—see Table 4.2 and Table 4.3. The following experiments were conducted on a batch of 50 images. A summary of the conducted experiments is listed in Table 4.1.

The logarithmic scale was used because according to our preliminary exploration of the hyperparameter space conducted in Chapter 3, where γ and ε values are most interesting near zero, AUC scores larger than one eventually stagnated.

LRP Composite	PF Sort
Decreasing-γ	MoRF
Decreasing-γ	LRF
Decreasing-γ	Random
Tutorial	MoRF
Tutorial	LRF
Tutorial	Random

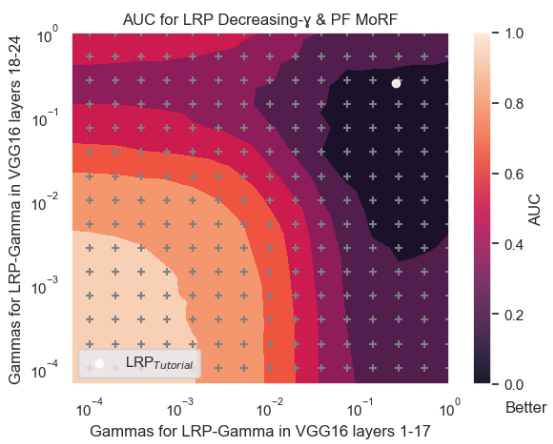
Table 4.1: Overview of Experiments.

4.1.1. Quantitative Evaluation of $\text{LRP}_{\text{Decreasing}-\gamma}$

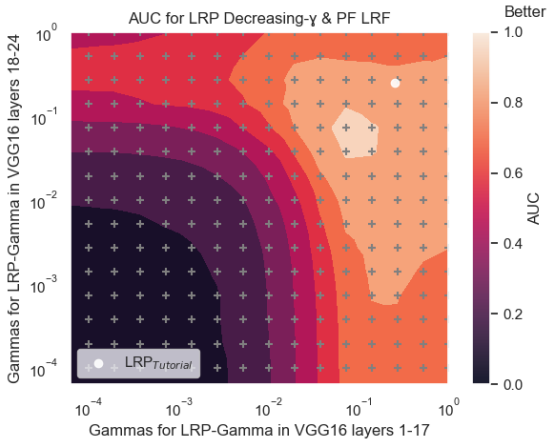
The hyperparameters for the composite evaluated in this section, $\text{LRP}_{\text{Decreasing}-\gamma}$, are defined in Table 4.2. In Figure 4.1a, permutations of 16 γ values for layers 1-17 and 18-24 of model VGG-16 were performed, a total of 256. These permutations are exemplified in Table 4.2. Given the results displayed in Figure 4.1b, we conducted further experiments on a reduced hyperparameter space to evaluate in-depth the discontinuity of AUC scores in the range between 0 and 0.2; the resulting AUC scores are shown in Figure 4.1d.

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
2-10	LRP_{γ}	$\gamma = 0.5$
11-17	LRP_{γ}	$\gamma \in \{0, 10^{-4}, \ldots, 10^{0}\}$
18-24	LRP_{γ}	$\gamma \in \{0, 10^{-4}, \ldots, 10^{0}\}$
25-31	LRP_{γ}	$\gamma = 0$

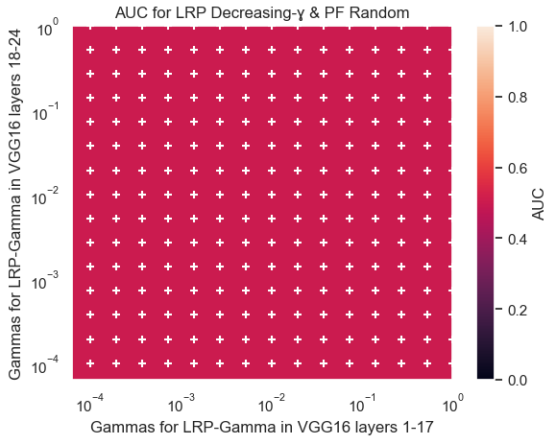
Table 4.2: Hyperparameter Space for $\text{LRP}_{\text{Decreasing}-\gamma}$.



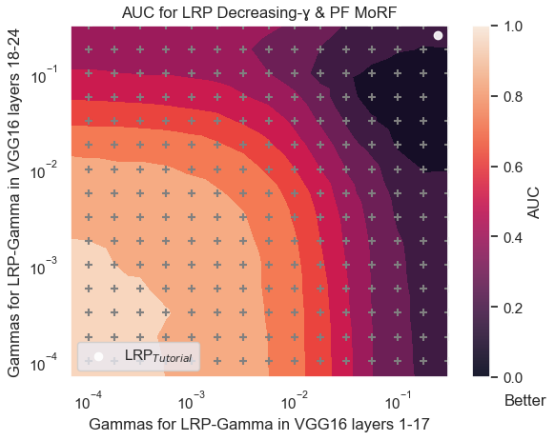
(a) AUC scores of $\text{LRP}_{\text{Decreasing}-\gamma}$, PF_{MoRF}



(b) AUC scores of $\text{LRP}_{\text{Decreasing}-\gamma}$, PF_{LRF}



(c) AUC Scores of $\text{LRP}_{\text{Decreasing}-\gamma}$, $\text{PF}_{\text{Random}}$



(d) AUC Scores for $\text{LRP}_{\text{Decreasing}-\gamma}$, PF_{MoRF}; Reduced Hyperparameter Space

Figure 4.1: Results for $\text{LRP}_{\text{Decreasing}-\gamma}$.

4.1.2. Quantitative Evaluation of $\text{LRP}_{\text{Tutorial}}$

The hyperparameter space explored for $\text{LRP}_{\text{Tutorial}}$ is summarized in Table 4.3. The results of the quantitative evaluation are depicted in Figure 4.2. Similarly to Subsection 4.1.1, the optimal hyperparameter values are closest to zero for γ and ε—see Figure 4.2a and Figure 4.2b. We notice that the original $\text{LRP}_{\text{Tutorial}}$ composite is relatively close to the optima shown by the evaluation framework and, in certain cases, it corresponds to the optimal explanation measured by AUC score.

4.2. Qualitative Results

In this section we visualize a sample from each experiment to compare the results qualitatively. The sample is taken from experiment number 140 out of the total 256; the number was chosen semi-randomly, purposely avoiding the ones where the hyperparameter is set to zero.

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
1-16	LRP_γ	$\gamma \in \{0, 10^{-4}, \ldots, 10^0\}$
17-30	LRP_ε	$\varepsilon \in \{0, 10^{-4}, \ldots, 10^0\}$
31-n	LRP_0	

Table 4.3: Hyperparameter Space for $\text{LRP}_{\text{Tutorial}}$.

Then, we will plot the optimal values from Section 4.1 in Figure 4.3. The hyperparameters corresponding to experiment number 140 are 0.01 and ≈ 0.138 (rounded). The respective composites tested are shown in Table 4.4 and Table 4.5.

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
2-10	LRP_γ	$\gamma = 0.5$
11-17	LRP_γ	$\gamma = 0.01$
18-24	LRP_γ	$\gamma \approx 0.138$
25-31	LRP_γ	$\gamma = 0$

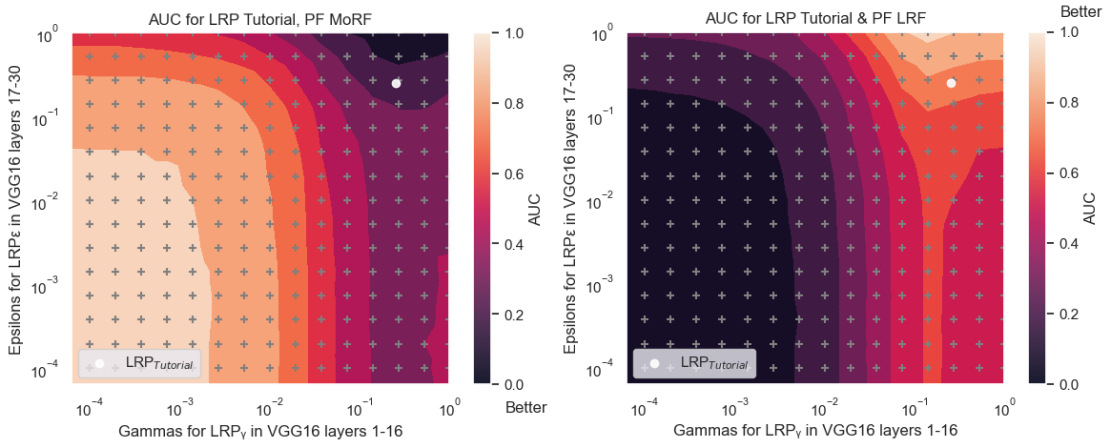
Table 4.4: Hyperparameters of $\text{LRP}_{\text{Decreasing}-\gamma}$ for Qualitative Comparison

Layer index	Rule	Hyperparameters
0 (pixel)	z^B	low, high
1-16	LRP_γ	$\gamma \in \{0, 10^{-4}, \ldots, 10^0\}$
17-30	LRP_ε	$\varepsilon \in \{0, 10^{-4}, \ldots, 10^0\}$
31-n	LRP_0	

Table 4.5: Hyperparameters of $\text{LRP}_{\text{Tutorial}}$ for Qualitative Comparison.

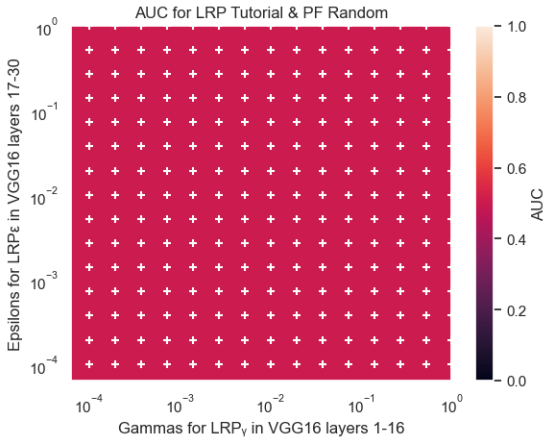
4.3. Discussion

The contour plots for $\text{LRP}_{\text{Decreasing}-\gamma}$ and PF_{MoRF} shown in Figure 4.1a meet our expectations by depicting that optimal hyperparameters according to our framework are found near zero. For $\text{LRP}_{\text{Tutorial}}$ and PF_{MoRF}, Figure 4.2a reveals that optimal hyperparameters regarding this composite are roughly $0.1 \leq \gamma \leq 1$ and $0.4 \leq \varepsilon \leq 1$. This confirms the hypothesis that LRP_γ



(a) AUC scores of $\text{LRP}_{\text{Tutorial}}$, PF_{MoRF}

(b) AUC scores of $\text{LRP}_{\text{Tutorial}}$, PF_{LRF}



(c) AUC scores of $\text{LRP}_{\text{Tutorial}}$, $\text{PF}_{\text{Random}}$

Figure 4.2: Results for $\text{LRP}_{\text{Tutorial}}$.

provides a measurable improvement over LRP_0, which is equivalent to $\text{LRP}_{\gamma=0}$. According to Figure 4.1d, the precise optimal values for γ are $0.05 \leq \gamma \leq 0.3$ for VGG-16 layers 1-17 and $0.01 \leq \gamma \leq 0.2$ for layers 18-24.

Plots for $\text{PF}_{\text{Random}}$ are constant, in line with the hypothesis that the choice of hyperparameters is independent of the performance of $\text{PF}_{\text{Random}}$—see Figure 4.1c and Figure 4.2c. Conversely, Figure 4.1b shows that optimal hyperparameters for γ are closest to zero. By definition, γ favors positive contributions, while PF_{LRF} negative ones. Thus, the optimal scores for PF_{LRF} are where the least positive contributions are preferred in favor of negative ones, which is when $\gamma = 0$. Similarly, for γ in $\text{LRP}_{\text{Tutorial}}$ in Figure 4.2b.

Figure 4.4 and Figure 4.5 show the penultimate steps of the perturbation process from the PF algorithm together with the original image and its heatmap generated by the LRP composites $\text{LRP}_{\text{Tutorial}}$ and $\text{LRP}_{\text{Decreasing}-\gamma}$. In the heatmap of Figure 4.5, positive relevances are concentrated around the eyes of the axolotl, which explains why the region around the eyes in Figure 4.4d is almost unperturbed by the inpainting process. In contrast to Figure 4.4c,

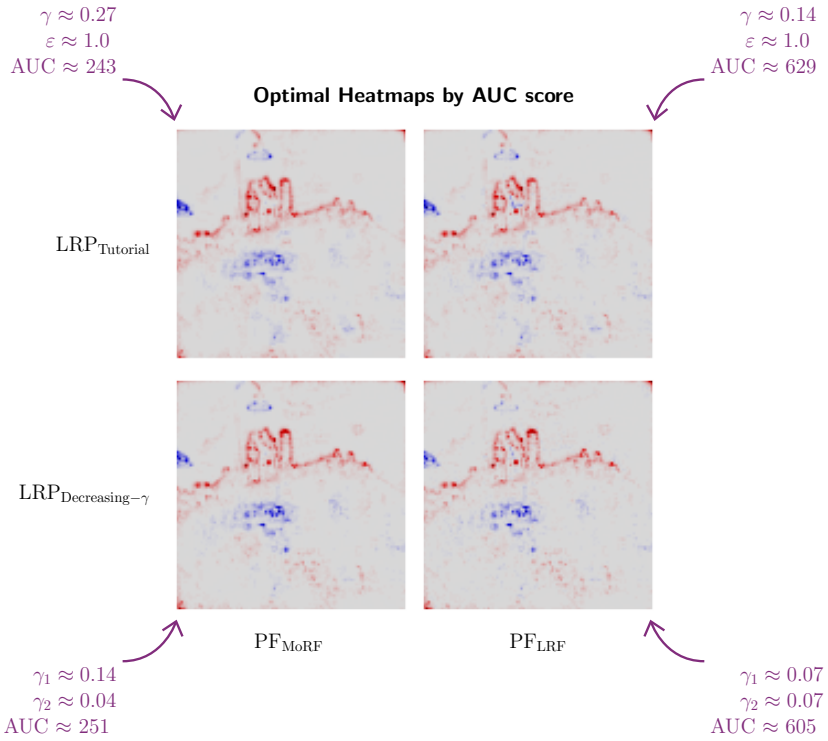


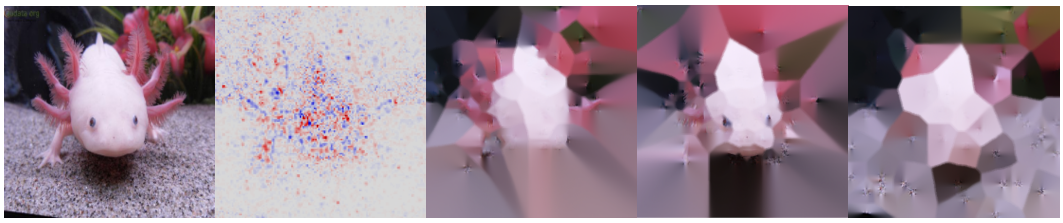
Figure 4.3: Optimal heatmaps according to AUC experiments

where the area around the eyes is the most perturbed.

Figure 4.5 shows scattered positive and negative relevances in the heatmap. As a result, the area around the axolotl's eyes in Figure 4.5d is more drastically perturbed compared to Figure 4.4d. Figure 4.4e and Figure 4.5e are practically indistinguishable due to the randomly inpainted regions. Figure 4.6 shows the distribution of AUC scores of all experiments. For PF_{LRF} and PF_{MoRF}, $\text{LRP}_{\text{Tutorial}}$ achieves the best quantitative results compared to $\text{LRP}_{\text{Decreasing}-\gamma}$. The qualitative differences between optimal heatmaps in Figure 4.3 are negligible; these optimal heatmaps are similar to the original $\text{LRP}_{\text{Tutorial}}$ heatmap (Montavon et al., 2019).

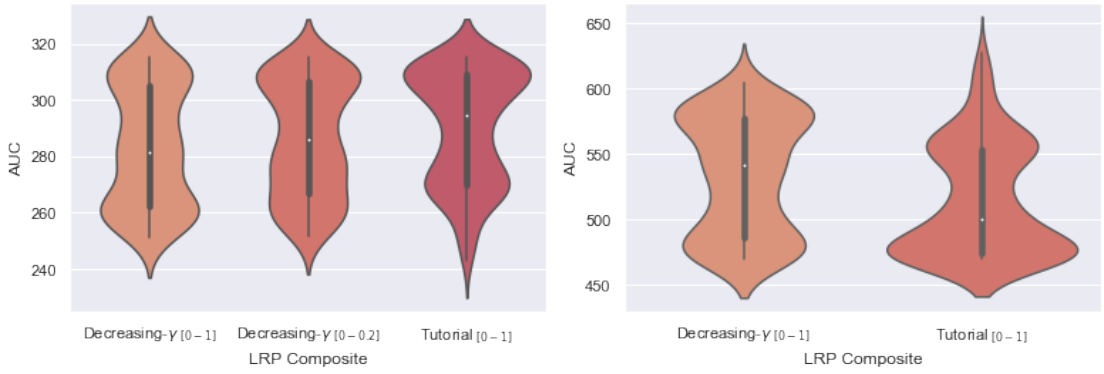
(a) Input **(b)** LRP Heatmap **(c)** PF_{MoRF} **(d)** PF_{LRF} **(e)** $\text{PF}_{\text{Random}}$

Figure 4.4: Qualitative Comparison of Perturbed Images for $\text{LRP}_{\text{Decreasing}-\gamma}$ as defined in Table 4.4.



(a) Input **(b)** LRP Heatmap **(c)** PF_{MoRF} **(d)** PF_{LRF} **(e)** PF_{Random}

Figure 4.5: Qualitative Comparison of Perturbed Images for $LRP_{Tutorial}$ as defined in Table 4.5.



(a) PF_{MoRF} **(b)** PF_{LRF}

Figure 4.6: AUC Scores of Experiments.

5. Conclusion

After reviewing state-of-the-art evaluation metrics for explanations of DNN as heatmaps, we have seen that there are several parameters which impact the quality and interpretation of the evaluation—e.g., choice of hyperparameters for LRP composite, choice of evaluation framework, implementations of LRP and PF, and visualization settings.

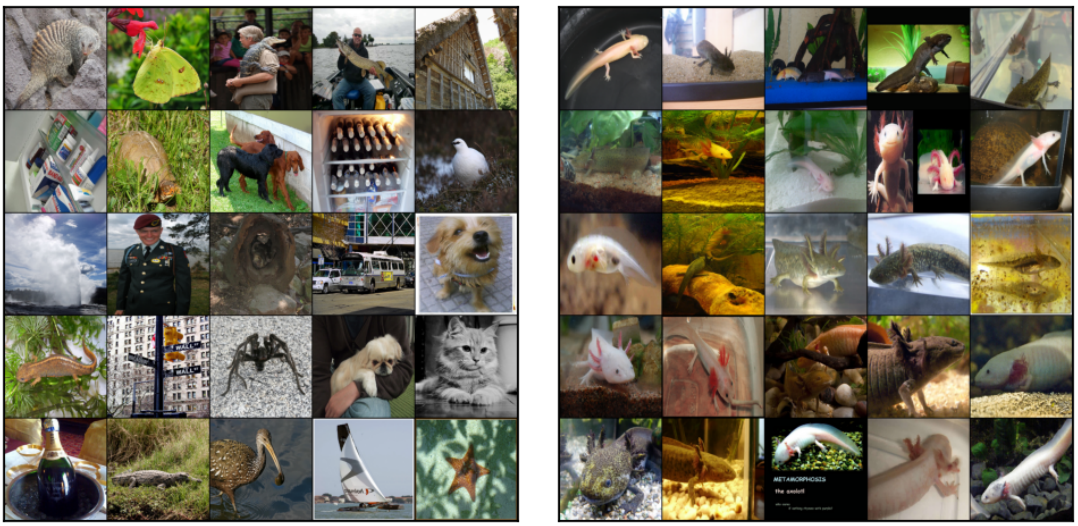
Depending on the visualization of these results, their interpretations can vary strongly. Therefore, we have demonstrated that there is no global optimum for heatmaps, the best representation is the one which most accurately matches the current use case. For instance, if aesthetic heatmaps are preferred, this directly depends on the opinion of the observer. If objectivity is preferred, it is possible to measure *best* according to a quantitative metric and an appropriate metric should be chosen—e.g., AOPC or AUC. Both are inherently different, hence, the dilemma on how to define *best* from the perspective of human interpretability. The choice of optimal LRP hyperparameters highly depends on the evaluation metrics, thus, no global optimum exists.

A. Dataset

ImageNet: The ImageNet Large Scale Visual Recognition Challenge 2012 dataset (Russakovsky et al., 2015)—also known as *ImageNet*—is available for download under https://www.image-net.org/challenges/LSVRC/2012/index.php and encompasses three tasks: (task 1) classification, (task 2) classification with localization, (task 3) fine-grained classification. The disjoint splits of the data set are: training, validation, and test. Samples of dataset shown in Figure A.1a.

The subsets originate from the ImageNet data set consisting of 10 million hand-labeled images with 10,000+ classes. The training data consists of 1.2 million images and 1,000 classes. The validation and test subsets consist of 150,000 images and 1,000 classes, from which 50,000 constitute the validation set. Each class in the validation set is limited to 50 samples in this set.

The validation split was used to conduct the experiments in Chapter 3 and Chapter 4 with the *axolotl* class—ground truth label 29. For the sake of completion, here is the synonym set (synset) of that class: axolotl, mud puppy, Ambystoma mexicanum. Samples of such class shown in Figure A.1b.



(a) Random samples **(b)** *axolotl* class samples

Figure A.1: ILSVRC 2012 images.

B. Additional Experiments

B.1. Qualitative Experiments

The following experiments in Figure B.1 were conducted using a different reference image than the castle image used in Chapter 3. As mentioned in Section 3.1 the changes in the heatmaps as ε increases are almost imperceptible, similarly as in Figure 3.3. The most noticeable changes are found when incrementing γ.

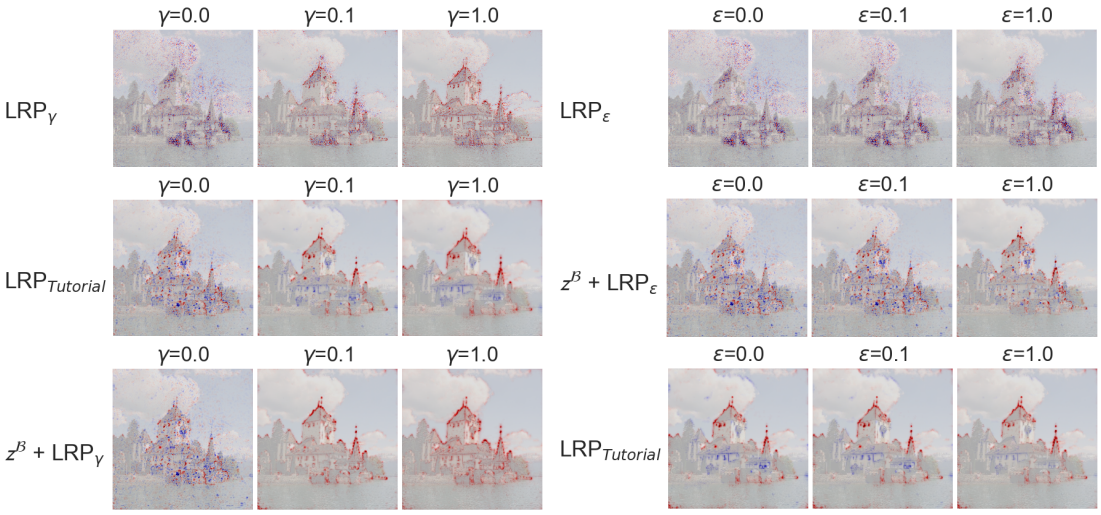


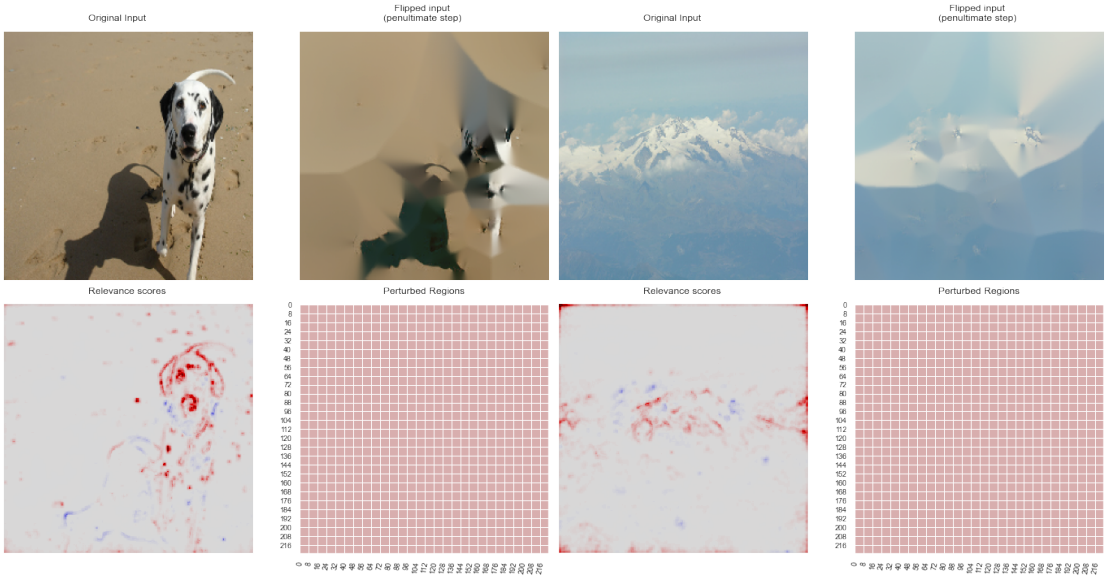
Figure B.1: Grid search for castle2 image.

B.2. Quantitative Experiments

In Section 3.3, we validate LRP-PF-AUC mainly based on the castle image and in Chapter 4 with *axolotl* images. The goal of this section is to reassure the reader that the methodology proposed is not limited to the few examples presented so far. The implementation of the framework will be validated by extending the scope to further classes of the ILSVRC 2012 dataset.

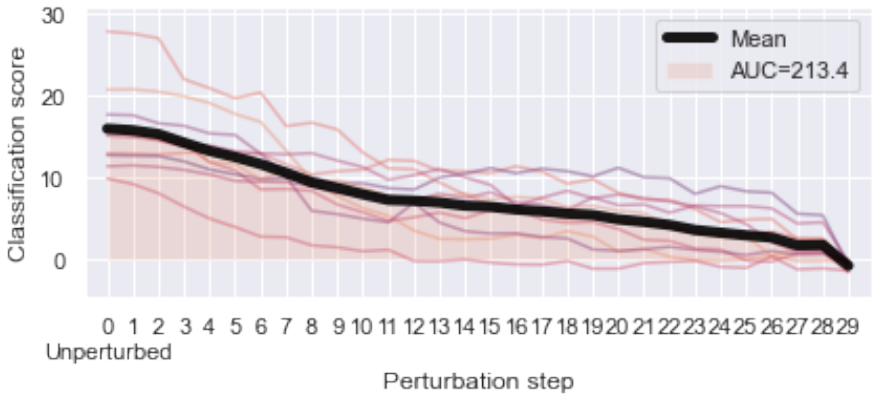
The perturbation curve displayed in Figure B.2c shows the expected prompt initial decline of the curve due to PF_{MoRF}. In Figure B.2a, we observe the negative evidence for *dalmatian* in the dog's shadow. Most likely, the images in the dataset on which the model was trained

included few examples of dog shadows. A relatively similar example is shown in Figure B.2b, where the negative class evidence for the *alp* class is due to the clouds. A possible assumption of the model is that the sky is predominantly cloudless in the Alps.



(a) Sample from *dalmatian* class

(b) Sample from *alp* class



(c) AUC scores for batch of 10 images from random ILSVRC 2012 classes

Figure B.2: $\text{LRP}_{\text{Tutorial}}$-$\text{PF}^{\text{Inpainting}}_{\text{MoRF}}$-AUC evaluation of image batch from random classes.

C. Implementing LRP

We will briefly compare the different approaches when implementing LRP to take into consideration possible impacts of the implementation on the experiments with the goal of differentiating them from the actual impact of hyperparameters on explanations, which is the top priority of this scope-limited research. We will also motivate design choices in our LRP implementation (Bermudez Schettino, 2022).

There are two known approaches: forward-hook or sequential implementations. The forward-hook implementation is more suitable for complex NN like ResNet50 with skip connections. The sequential implementation is the most common one; it involves using a forward-backward loop and it is more intuitive for architectures such as VGG-16. Here we will give an overview of how they differ from each other. In this chapter we will assume the models are already in canonical form; canonization was discussed in Subsection 2.1.4.

Implementing LRP in PyTorch is recommended due to its popularity (Kohlbrenner et al., 2020). For this thesis, LRP was implemented from scratch despite the alternatives available (Kokhlikyan et al., 2020; Hedström et al., 2022; Alber et al., 2018) to provide an end-to-end explanation and evaluation framework of NN results, as well as to develop a better understanding of the algorithms at hand.

C.1. Sequential Algorithm

The following algorithms are the implementation of equation (2.3), which corresponds to $\text{LRP}_0/\varepsilon/\gamma$.

C.1.1. Definition A - $\text{LRP}_0/\varepsilon/\gamma$

$$\forall_k : z_k = \varepsilon + \sum_{0,j} a_j \cdot \rho(w_{jk}) \qquad \text{1. Forward pass} \qquad \text{(C.1)}$$

$$\forall_k : s_k = \frac{R_k}{z_k} \qquad \text{2. Element-wise division} \qquad \text{(C.2)}$$

$$\forall_j : c_j = \sum_k \rho\left(w_{jk}\right) \cdot s_k \qquad \text{3. Backward pass} \qquad \text{(C.3)}$$

$$\forall_j : R_j = a_j \cdot c_j \qquad \text{4. Element-wise product} \qquad \text{(C.4)}$$

Operation in equation (C.1) is done on a copy of the original layer. Equation (C.2) is equivalent to:

$$c_j = \left[\nabla \left(\sum_k z_k(a) \cdot s_k \right) \right]_j \tag{C.5}$$

where $a = (a_j)_j$, z_k is a function, s_k is constant.

Special case - LRP$_\gamma$

Forward pass:

$$\forall_k : z_k = \sum_{0,j} a_j \cdot \rho\left(w_{jk}\right) \tag{C.6}$$
$$= \sum_{0,j} a_j \cdot \left(w_{jk} + \gamma w_{jk}^+\right) \tag{C.7}$$
$$\tag{C.8}$$
$$\text{where } \rho(w_{jk}) = w_{jk} + \gamma w_{jk}^+ \tag{C.9}$$

Backward pass:

$$\forall_j : c_j = \sum_k \rho\left(w_{jk}\right) \cdot s_k \tag{C.10}$$
$$\sum_k \left(w_{jk} + \gamma w_{jk}^+\right) \frac{a_k}{z_k} c_k \tag{C.11}$$
$$= \sum_k \frac{\partial z_k}{\partial a_j} \cdot \frac{\partial a_k}{\partial z_k} \cdot c_k \tag{C.12}$$
$$= \sum_k \frac{\partial z_k \left[\frac{a_k}{z_k}\right]_{cst.} c_k}{\partial a_j} \tag{C.13}$$
$$s_k = \frac{R_k}{z_k} = \frac{a_k c_k}{z_k} \tag{C.14}$$
$$R_k = a_k c_k \tag{C.15}$$

Replace a_k by $z_k \cdot \left[\frac{a_k}{z_k}\right]_{cst.}$ in the forward pass. Valid for linear and convolution layers to support automatic differentiation. This rule definition is only valid for linear and convolutional layers.

C.1.2. Definition B - LRP$_0$/ε/γ

The above algorithm is equivalent to the following one but with different notation. Step (C.16) is a forward evaluation of a copy of the original layer, similar as in (C.1).

$$z = \varepsilon + f_l^{\rho}\left(a^{(l-1)}\right) \quad \text{1. Forward pass} \tag{C.16}$$
$$s = R^{(l)} \oslash z \quad \text{2. Element-wise division} \tag{C.17}$$
$$c = \nabla\langle z, [s]_{cst.}\rangle \quad \text{3. Backward pass} \tag{C.18}$$
$$R^{(l-1)} = a \odot c \quad \text{4. Element-wise product} \tag{C.19}$$
$$\textbf{return } R^{(l-1)} \tag{C.20}$$

C.1.3. Definition B - z^B

$$z = f_1(x) + f_1^+(l) - f_1^-(h) \tag{C.21}$$
$$s = R^{(1)} / z \tag{C.22}$$
$$c = \nabla_{x,l,h}\langle z, [s]_{cst.}\rangle \tag{C.23}$$
$$R^{(0)} = x \odot c_1 + l \odot c_2 + h \odot c_3 \tag{C.24}$$
$$\textbf{return } R^{(0)} \tag{C.25}$$

C.2. Forward-Hook Algorithm

The forward-hook implementation of the LRP algorithm for DRN (Samek et al., 2021) has advantages over the sequential one, as it can be implemented seamlessly for complex networks—e.g., ResNet50 with skip connections.

$$\begin{aligned} &y = f(\boldsymbol{x}, \boldsymbol{l}, \boldsymbol{h}) \\ &c_1, c_2, c_3 = \hat{\nabla} y \\ &R = x \odot c_1 = l \odot c_2 + h \odot c_3 \\ &\textbf{return } R \end{aligned} \tag{C.26}$$

Equation C.26: Global LRP Computation (Forward-hook) (Samek et al., 2020).

There are multiple strategies to handle (i.e., propagate relevance through) MaxPooling2D layers in forward-hook implementations: winner-takes-all and sum-pooling rules. Both approaches are not equivalent. Max-pooling in forward-hook already handled automatically by *autograd*.

The following definitions only apply for $\text{LRP}_0/\varepsilon/\gamma$ in DRN.

C.2.1. First Layer

$$\begin{aligned} z &= f_1(\boldsymbol{x}) - f_1^+(\boldsymbol{l}) - f_1^-(\boldsymbol{h}) \\ \text{return } & z \odot \left[f_1(x) \oslash z\right]_{cst.} \end{aligned} \tag{C.28}$$

$$\begin{aligned} z &= \varepsilon + f_l^{\rho}\left(\boldsymbol{a}^{(l-1)}\right) \\ \text{return } & z \odot \left[f_l\left(\boldsymbol{a}^{(l-1)} \oslash z\right)\right]_{cst.} \end{aligned} \tag{C.27}$$

Equation C.27: Factor required to be stabilized in implementation for forward-hook implementation of $\text{LRP}_0/\varepsilon/\gamma$ for intermediate layers. See equation (C.29).

C.2.2. Intermediate Layers

$$\begin{aligned} z &= \varepsilon + f_l^{\rho}\left(\boldsymbol{a}^{(l-1)}\right) \\ \text{return } & z \odot \left[f_l\left(\boldsymbol{a}^{(l-1)} \oslash z\right)\right]_{cst.} \end{aligned} \tag{C.29}$$

f_l^{ρ} is a forward pass on a copy layer where parameters are processed by the ρ function.

Bibliography

Chirag Agarwal and Anh Nguyen. Explaining Image Classifiers by Removing Input Features Using Generative Models. In Hiroshi Ishikawa, Cheng-Lin Liu, Tomas Pajdla, and Jianbo Shi, editors, Computer Vision – ACCV 2020, volume 12627, pages 101–118. Springer International Publishing, Cham, 2021. ISBN 978-3-030-69543-9 978-3-030-69544-6. doi: 10.1007/978-3-030-69544-6_7. URL http://link.springer.com/10.1007/978-3-030-69544-6_7. Series Title: Lecture Notes in Computer Science.

Maximilian Alber, Sebastian Lapuschkin, Philipp Seegerer, Miriam Hägele, Kristof T. Schütt, Grégoire Montavon, Wojciech Samek, Klaus-Robert Müller, Sven Dähne, and Pieter-Jan Kindermans. iNNvestigate neural networks!, August 2018. URL http://arxiv.org/abs/1808.04260. Number: arXiv:1808.04260 arXiv:1808.04260 [cs, stat].

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. XAI for Transformers: Better Explanations through Conservative Propagation, February 2022. URL http://arxiv.org/abs/2202.07304. Number: arXiv:2202.07304 arXiv:2202.07304 [cs].

Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for Dataset-wide XAI: From Local Explanations to Global Insights with Zennit, CoRelAy, and ViRelAy. CoRR, abs/2106.13200, 2021.

Christopher J Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. Information Fusion, 77:261–295, 2022. Publisher: Elsevier.

Léo Andéol, Yusei Kawakami, Yuichiro Wada, Takafumi Kanamori, Klaus-Robert Müller, and Grégoire Montavon. Learning Domain Invariant Representations by Joint Wasserstein Distance Minimization. arXiv preprint arXiv:2106.04923, 2021.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. PLOS ONE, 10(7):e0130140, July 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0130140. URL https://dx.plos.org/10.1371/journal.pone.0130140.

Rodrigo Bermudez Schettino. Rodrigo Bermudez Schettino / LRP tutorial · GitLab, March 2022. URL https://git.tu-berlin.de/rodrigobdz/lrp-tutorial.

Alexander Binder. Notes on Canonization for Resnets and Densenets. page 21, July 2020.

Kirill Bykov, Mayukh Deb, Dennis Grinwald, Klaus-Robert Müller, and Marina M.-C. Höhne. DORA: Exploring outlier representations in Deep Neural Networks. Technical Report arXiv:2206.04530, arXiv, June 2022. URL http://arxiv.org/abs/2206.04530. arXiv:2206.04530 [cs, stat] type: article.

Oliver Eberle, Jochen Büttner, Florian Kräutli, Klaus-Robert Müller, Matteo Valleriani, and Grégoire Montavon. Building and Interpreting Deep Similarity Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(3):1149–1161, March 2022. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2020.3020738. URL http://arxiv.org/abs/2003.05431. arXiv:2003.05431 [cs, stat].

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. University of Montreal, 1341(3):1, 2009.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE international conference on computer vision, pages 3429–3437, 2017.

Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations. 2022. _eprint: 2202.06861.

Haris Iqbal. HarisIqbal88/PlotNeuralNet v1.0.0, December 2018. URL https://doi.org/10.5281/zenodo.2526396.

Jacob Kauffmann, Malte Esders, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. From clustering to cluster explanations via neural networks. arXiv preprint arXiv:1906.07633, 2019.

Jacob Kauffmann, Lukas Ruff, Grégoire Montavon, and Klaus-Robert Müller. The clever Hans effect in anomaly detection. arXiv preprint arXiv:2006.10609, 2020.

Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. Towards Best Practice in Explaining Neural Network Decisions with LRP, July 2020. URL http://arxiv.org/abs/1910.09840. Number: arXiv:1910.09840 arXiv:1910.09840 [cs, stat].

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for PyTorch, 2020. _eprint: 2009.07896.

Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. The LRP Toolbox for Artificial Neural Networks. Journal of Machine Learning Research, 17(114):1–5, 2016. URL http://jmlr.org/papers/v17/15-618.html.

Sebastian Lapuschkin, Alexander Binder, Klaus-Robert Müller, and Wojciech Samek. Understanding and Comparing Deep Neural Networks for Age and Gender Classification, August 2017. URL http://arxiv.org/abs/1708.07689. Number: arXiv:1708.07689 arXiv:1708.07689 [cs, stat].

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. Nature Communications, 10(1):1096, December 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-08987-4. URL http://www.nature.com/articles/s41467-019-08987-4.

Simon Letzgus, Patrick Wagner, Jonas Lederer, Wojciech Samek, Klaus-Robert Müller, and Gregoire Montavon. Toward Explainable AI for Regression Models. Technical Report arXiv:2112.11407, arXiv, December 2021. URL http://arxiv.org/abs/2112.11407. arXiv:2112.11407 [cs, stat] type: article.

Jan Macdonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. Explaining neural network decisions is hard. In XXAI Workshop, 37th ICML, 2020.

Grégoire Montavon. gmontavon / LRP tutorial · GitLab, December 2021. URL https://git.tu-berlin.de/gmontavon/lrp-tutorial.

Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining NonLinear Classification Decisions with Deep Taylor Decomposition. Pattern Recognition, 65:211–222, May 2017. ISSN 00313203. doi: 10.1016/j.patcog.2016.11.008. URL http://arxiv.org/abs/1512.02479. arXiv:1512.02479 [cs, stat].

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for Interpreting and Understanding Deep Neural Networks. Digital Signal Processing, 73:1–15, February 2018. ISSN 10512004. doi: 10.1016/j.dsp.2017.10.011. URL http://arxiv.org/abs/1706.07979. arXiv:1706.07979 [cs, stat].

Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-Wise Relevance Propagation: An Overview. In Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors, Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pages 193–209. Springer International Publishing, Cham, 2019. ISBN 978-3-030-28954-6. doi: 10.1007/978-3-030-28954-6_10. URL https://doi.org/10.1007/978-3-030-28954-6_10.

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), 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Wojciech Samek. Meta-Explanations, Interpretable Clustering & Other Recent Developments, November 2019.

Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In Explainable AI: interpreting, explaining and visualizing deep learning, pages 5–22. Springer, 2019.

Wojciech Samek, Alexander Binder, Gregoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Muller. Evaluating the Visualization of What a Deep Neural Network Has Learned. IEEE Transactions on Neural Networks and Learning Systems, 28(11):2660–2673, November 2017. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2016.2599820. URL https://ieeexplore.ieee.org/document/7552539/.

Wojciech Samek, Grégoire Montavon, Klaus-Robert Müller, TU Berlin, and TU Berlin. EMBC Tutorial on Interpretable and Transparent Deep Learning. page 56, 2019.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Toward interpretable machine learning: Transparent deep neural networks and beyond. 2020.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. Proceedings of the IEEE, 109(3):247–278, March 2021. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2021.3060483. URL http://arxiv.org/abs/2003.07631. arXiv:2003.07631 [cs, stat].

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713, 2016.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. Technical Report arXiv:1409.1556, arXiv, April 2015. URL http://arxiv.org/abs/1409.1556. arXiv:1409.1556 [cs] type: article.

William Swartout and Johanna Moore. Explanation in Second Generation Expert Systems. January 1993. doi: 10.1007/978-3-642-77927-5_24. Pages: 585.

Ole Tange. GNU Parallel 20220622 ('Bongbong'), June 2022. URL https://doi.org/10.5281/zenodo.6682930.

Alexandru Telea. An image inpainting technique based on the fast marching method. Journal of graphics tools, 9(1):23–34, 2004. Publisher: Taylor & Francis.

Erico Tjoa and Cuntai Guan. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. IEEE Transactions on Neural Networks and Learning Systems, 32(11):4793–4813, November 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.3027314. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.