Comparison of Explainable AI Methods for Object Detection

Maximilian Dreyer (Matrikel-Nr. 806092)
Research Seminar of the Machine Learning Group by Prof. Tobias Scheffer* (University of Potsdam)
Supervised by Paul Prasse†(University of Potsdam)
and Sebastian Lapuschkin‡(Fraunhofer HHI Berlin)

September 13, 2021

Abstract

Deep neural networks are often seen as black-boxes, producing an output without explanation. Explainable Artificial Intelligence (XAI) methods improve our understanding of these often opaque networks. So called post-hoc methods have most often been applied to classification tasks in order to determine the relevance of input variables. This work focuses on implementing the post-hoc XAI methods LRP, DeepLIFT, LIME and SHAP for the task of object detection using the Single Shot Detector (SSD) model architecture. All methods are further compared regarding human interpretability, faithfulness and applicability. Here, the necessity of a robust propagation rule for backpropagation-based methods is shown and discussed.

1 Introduction

In the last decade, deep neural networks have become a powerful tool in a variety of areas like science, business and engineering [Pan15]. Although the application of deep learning models can lead to highly accurate predictions, the decision process is generally opaque due to a large parameter space (millions of parameters). In application areas like automated driving or disease detection, where predictions can have a large impact on human lives, transparency and traceability is crucial [Hol+19]. In fact, explanations of reasoning mechanisms for such applications are already legally regulated in the European Union by the General Data Protection Regulation [16].

Various XAI techniques and ideas have been developed and applied in the past to increase transparency [TG20]. On one hand, models that are interpretable by design can be used, or the architecture can be changed in order to improve interpretability. On the other hand, oftentimes opaque standard architectures (e.g, ResNet [He+16] or VGG [SZ14]) with pre-trained parameters are used as starting points. Then, explanations can also be generated by means of external techniques (post-hoc explainability). These two different views can also be seen as solving the black-box problem against explaining the black-box [Gui+18].

In the case of deep neural networks, architectures are in general not interpretable, but can be changed in order to improve transparency (e.g. by adding/changing layers). One example is the Concept-Whitening-Layer, introduced by Chen et al. [CBR20], which can be used to align the latent space in order to better investigate the learned concepts of a model.

Regarding post-hoc explainability for deep learning models, feature relevance explanation methods are widely used [Arr+20]. These

*scheffer@cs.uni-potsdam.de
†prasse@cs.uni-potsdam.de
‡sebastian.lapuschkin@hhi.fraunhofer.de

methods clarify the inner functioning of a model by computing a relevance score for the input.

This work compares four feature relevance explanation methods: LRP [Bac+15], DeepLIFT [SGK17], SHAP [LL17] and LIME [RSG16].
In the first two sections, the methods are introduced and applied to the domain of bounding box detection using the Single Shot Detector (SSD) model [Liu+16]. Thereafter, the four methods are compared regarding faithfulness, human interpretability and applicability followed by a conclusion. In order to evaluate the faithfulness of explanations, localization and pixel-flipping experiments are performed and evaluated. For applicability and human interpretability, run time and heatmap entropy is measured, respectively.

1.1 Related Work

In order to compare and evaluate explanation heatmaps, several techniques have been developed in the past years. They can be categorized into "sanity" checks, localization and insertion/deletion experiments, human interpretability assessments and applicability tests.

Sanity Check Model parameter randomization can be used to check the "sanity" of explanations, i.e. whether the methods adhere to and are sensitive to parameters of the model [Ade+18; HLA21]. The idea is that the change in an explanation should be correlated to parameter randomization, because the output of the model changes as well.

Localization Localization techniques check whether the relevant regions of a heatmap coincide with the classified object in the image. Fong et al. propose to use intersection over union (IOU) between relevant pixel areas and the classified object [FV17]. Here, bounding boxes for the object as well as for the relevant pixels are predicted. Alternatively, Zhang et al. investigate in a pointing game whether the most relevant pixel is a pixel of the object being classified [Zha+18].

Insertion and Deletion Perturbation of the most relevant pixels should have a negative effect on the prediction score. The authors of [Bac+15] investigate the faithfulness of explanations by pixel flipping. Therefore, Montavon et al. propose to quantify the faithfulness by calculating the area under curve (AUC) [MSM18; Sam+16]. Contrary, it is also possible to start with an empty input, insert the most relevant pixels and analyze the increase in the network's prediction abilities [PDS18; HLA21].

Human Interpretability Samek et al. [Sam+16; Sam+21] propose to quantify human interpretability in terms of the amount of information contained in the heatmap for the image classification setting. They measure the information amount via the file size, as a high associated file size is more likely to contain complex features.

Applicability Regarding applicability, it can be generally compared whether methods can be applied to all models (model-agnostic) or only specific models (model-specific). Further, the run time of a method can be crucial for the application, e.g., when a lot of explanations are necessary. Thus, several works compare the mean time it takes to generate an explanation [HLA21; Sam+21].

2 Post-hoc XAI methods

In the following section, the four post-hoc XAI methods LRP, DeepLIFT, LIME and SHAP are introduced.

2.1 LRP

Layer-wise Relevance Propagation (LRP) is a method for explaining neural networks using backpropagation of attributions. LRP quantifies the relevance of the input as well as all intermediate neurons for the output of the model.

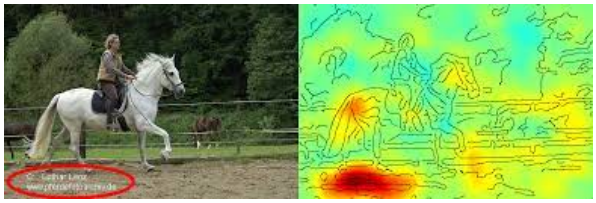

Figure 1: Image classification for the class "horse": XAI method explanations (here from LRP) show that the horse is predicted because of the copyright text. This suggests that the training data should be cleaned of watermarks in order for the network to generalize better. Source: [Lap+16]

Given an image as input, the explanation can be visualized as a heatmap, as is possible for all presented XAI methods (see Figure 1). This offers a visualization of which pixels contributed positively and which negatively to the decision of the model.

The idea of LRP is to propagate the output $f(\mathbf{x})$ backward through the network by using specific propagation rules. The propagation is subject to a conservation principle, where the relevance attributed to a neuron will be passed completely to all lower level neurons.

The general LRP propagation rule is given for neurons j and k at two consecutive layers of the neural network. An illustration of the flow of relevance across layers in the network is depicted in Figure 2. Propagation of relevances R_k to its in the architecture preceding neuron j is achieved by applying:

$$R_j = \sum_k \frac{z_{jk}}{\sum_l z_{lk}} R_k \tag{2.1}$$

Here, z_{jk} models the extent to which neuron j has contributed to activate neuron k. The denominator ensures the conservation property $\sum_j R_j = \sum_k R_k$ — meaning the total relevance is conserved for two consecutive layers of the model.

For linear layers, followed by a rectifier nonlinearity (ReLU), which is often used in neural networks such as VGG-16 [SZ14], the basic propagation rule is given by Equation (2.1) using $z_{jk} = a_j w_{jk}$ with activation a_j of neuron j

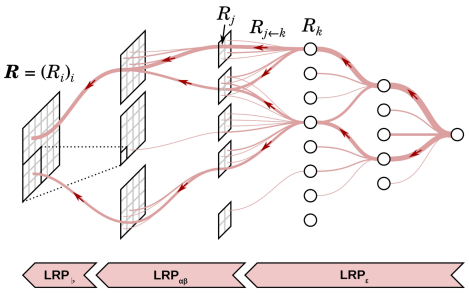

Figure 2: Relevance is distributed from the output to lower level neurons by applying LRP. Different rules are applied for the top, center and bottom part of the network. Source: [MSM18]

and weights w_{jk}. Here, the sum in the denominator also runs over the bias w_{0k}.

The basic LRP rule works as long as the denominator is not close to zero. In general, one can adapt the propagation rule to improve the explanations (e.g. by adding a small constant to the denominator to improve stability). In fact, it is good practice to use different rules for specific parts of the network (see [Mon+19] for an overview of rules).

2.2 DeepLIFT

Deep Learning Important FeaTures (DeepLIFT) [SGK17] is, like LRP, a method that uses attribution backpropagation. Crucial for the approach is a reference input, which should represent the state without any information. The main idea is to compare the activation of each neuron to its reference activation and to assign contribution scores according to the differences.

Let t be the activation of a target output neuron. The difference in activation compared to the reference activation t_0 is then given by $\Delta t = t - t_0$. DeepLIFT subsequently assigns a contribution score $C_{\Delta a_i \Delta t}$ to each input neuron's change in activation Δa_i, such that:

$$\Delta t = \sum_i C_{\Delta a_i \Delta t} \,.$$

The difference in activation in the target neu-

ron is assigned to the input neurons. This can be rewritten in a way that resembles the idea of partial derivatives. Writing $m_{\Delta a_i \Delta t} = \frac{C_{\Delta a_i \Delta t}}{\Delta a_i}$ leads to

$$\Delta t = \sum_i m_{\Delta a_i \Delta t} \Delta a_i$$

with the multiplier $m_{\Delta a_i \Delta t}$ similar to $\frac{\partial y}{\partial a_i}$ in the sense of finite differences. Having intermediate neurons y_j, one further receives via the chain rule

$$m_{\Delta a_i \Delta t} = \sum_j m_{\Delta a_i \Delta y_j} m_{\Delta y_j \Delta t}$$

For a simple linear layer with output $y = \sum_i w_i a_i$ given by weights w_i and inputs a_i, one gets the Linear rule

$$m_{\Delta a_i \Delta y} = w_i$$

and for a nonlinear layer the Rescale rule

$$m_{\Delta a \Delta y} = \frac{\Delta y}{\Delta a}\,. \tag{2.2}$$

There is also the RevealCancel Rule that explicitly handles positive and negative contributions and is an approximation to Shapely Values (see Section 2.4).

2.3 LIME

Local Interpretable Model- Agnostic Explanations (LIME) [RSG16] tries to find importance of contiguous superpixels (patches of pixels) in an input image towards the output of a neural network. The idea is to replace the decision function by a local surrogate model that is structured in a self-explanatory way (like a linear model).

In order to have a surrogate model g of model class G, that is explainable, it should be as simple as possible. This is represented by a complexity measure Ω. On the other hand, g should be faithful. The loss term $\mathbf{L}(f, g, \pi_x)$ represents the unfaithfulness of g regarding the network f in the locality defined by π_x.

The explanation is then given by minimizing

$$\xi(x) = \operatorname*{argmin}_{g \in G} \mathbf{L}(f, g, \pi_x) + \Omega(g)\,.$$

In a simple case, G might be the class of linear functions, π_x a Gaussian distribution and $\mathbf{L}$ incorporates a squared loss. The minimization takes place via weighted sampling around the input data.

2.4 SHAP

SHapley Additive exPlanation (SHAP) [LL17] is an additive feature attribution method. Perturbations of the input are used to compute attributions, based on the concept of Shapley Values from cooperative game theory.

Given a reference (or baseline) input, a random permutation of the input features is added one-by-one. After each step, the network is evaluated. The output difference after adding each feature corresponds to its attribution. But the ordering in which the features are added is important for nonlinear functions. Thus, these differences are averaged when repeating this process several times, each time choosing a new random permutation of the input features.

3 Application to Object Detection

In the following section, the introduced methods are applied to the task of object detection. Therefore, the model architecture and dataset are described first.

3.1 Model Architecture and Dataset

The architecture to be investigated is the Single-Shot Detector (SSD) [Liu+16]. The input of the model is an RGB-image of size 512×512 pixels. The output of the SSD network consists of bounding box coordinates with class scores. For the backbone, a VGG-16 [SZ14] is used as a feature extractor. As can be seen in Figure 3, layer after layer, the height and width of the intermediate input tensor becomes smaller. The bounding boxes are

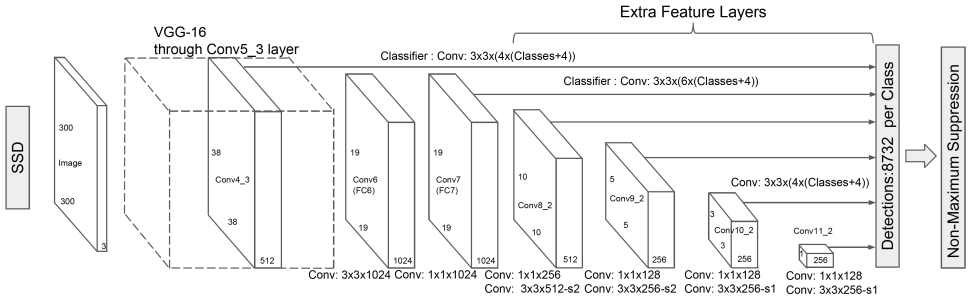

Figure 3: Model architecture of the Single Shot Detector (SSD) network. Source: [Liu+16]

predicted after each layer using a 3×3 convolution. Thus, at earlier stages, the size of the bounding boxes is smaller. The bounding box at the last stage (`Conv11_2`) covers the whole input image.

The SSD architecture with VGG-16 backbone consists of convolutions, ReLU activations and an ℓ_2-Norm layer. The ℓ_2-Norm layer transforms the input as

$$y_{cwh} = \alpha_c \frac{x_{cwh}}{\sqrt{\sum_i x_{iwh}^2}} \tag{3.1}$$

with weight $\alpha_c \in \mathbb{R}$ and dimensions for the channel (c), width (w) and height (h) of the input x. Thus, the ℓ_2-Norm normalizes and scales the input.

The evaluation is done on the Microsoft Common Objects in Context (COCO) dataset [Lin+14] which consists of 80 object categories like car, bicycle, human and dog. For each object of an image in the dataset, a segmentation mask as well as bounding box coordinates are given.

3.2 Adaption of Methods

The XAI methods DeepLIFT, LIME and SHAP are implemented using the XAI framework Captum [N+19] for PyTorch. Due to the limited implementation of LRP rules in Captum, the Zennit package [And+21] is used for LRP instead. In the following, all XAI methods are adapted to the task of object detection.

3.2.1 LRP

LRP has successfully been implemented for networks with convolutional layers and ReLU activations. The ℓ_2-Norm layer (see Equation (3.1)) is a special layer of the SSD model. In order to obtain optimal results, it is known that LRP rules need to be carefully aligned to the layers and their function within the model [Mon+19; Koh+20]. Simply using the gradient violates the conservation principle of LRP. In first order Taylor approximation one receives for a neuron y_{cwh} with a reference point $\hat{x}$ where $\hat{x}_{cwh} = 0$

$$y_{cwh} \approx \frac{\alpha_c}{\sqrt{\sum_{i \neq c} \hat{x}_{ihw}^2}} x_{cwh} \,. \tag{3.2}$$

Thus, in first order approximation, the ℓ_2-Norm is scaling the input x_{cwh} similarly to an activation. This leads to the approach of handling the ℓ_2-layer like an activation layer and passing the relevance of neuron x_{cwh} to y_{cwh}.

In order to achieve optimally interpretable heatmaps, different rules are tested (see appendix A for a list of all used rules). To further improve visualization, the $\flat$-rule [Bac+16] is applied for the first layer, meaning relevance is propagated to all pixels inside a neuron's receptive field. This way, the heatmap is becoming softer and features are more visible.

For deep neural networks, it has been shown that the gradient is noisy, partly due to gradient shattering [Bal+17]. This is why the ϵ-rule, which is similar to propagation of the gradient, results in noisy heatmaps (see Figure 4). Clearer heatmaps result from the z^+ and γ-

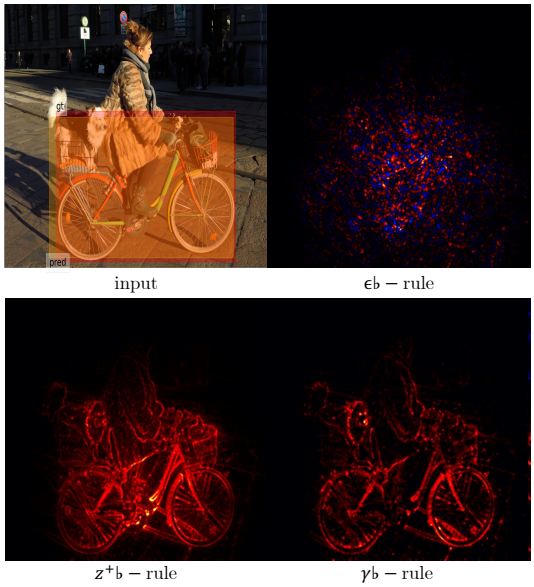

Figure 4: LRP explanations for bounding box and class of 'bicycle' using the ϵ, $\gamma(0.1)$, and z^+-rule in combination with the $\flat$-rule. In the input image, the bounding boxes for the ground truth (red) and prediction (yellow) are depicted.

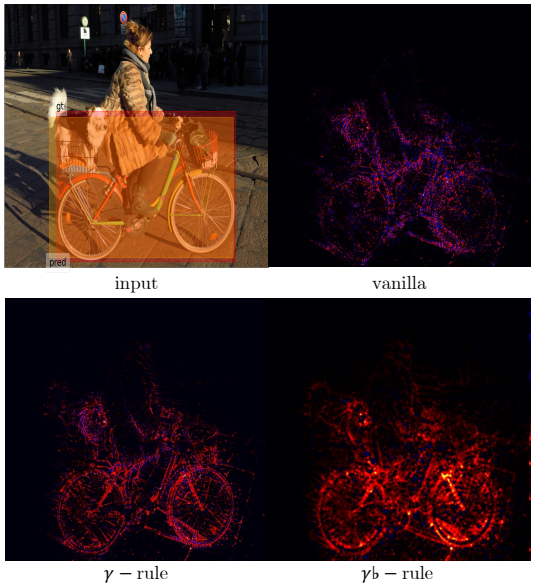

Figure 5: DeepLIFT explanations for the same setting as LRP in Figure 4. DeepLIFT without changes (vanilla) is noisy like LRP with the ϵ-rule, whereas applying the $\gamma(0.1)$ and $\flat$-rule leads to clearer and softer heatmaps.

rule, when applied to all convolutional layers. Compared to the z^+-rule, application of the γ-rule is in this case more sensible, because it differentiates between positive and negative contributions. As can be seen by comparing the heatmaps, the z^+-rule leads to more parts of the heatmap receiving relevance, as a lot of pixels contributed positively in some way to the output. The γ-rule, on the other hand, leads to more balanced heatmaps in terms of relevance and is thus used.

3.2.2 DeepLIFT

DeepLIFT supports convolutional and ReLU layers similar to LRP. Regarding the ℓ_2-Norm layer, the Rescale rule for nonlinear layers (2.2) is used.

As the authors of [SGK17] suggest for image data, a Gaussian blurred version of the input image is used as a reference sample. Therefore, a standard deviation of 12 corresponding to a kernel size of 80×80 is chosen in order to blur small as well as large features.

Using vanilla DeepLIFT, the heatmap is noisy, like using the LRP ϵ-rule. This can be seen by comparing Figure 4 and 5. In order to improve heatmap interpretability, the idea of the LRP γ-rule is used for DeepLIFT as well. In the following, for all convolutional layers, the positive weights w are favored by a factor of $1+\gamma$, meaning $w \longrightarrow (1+\gamma)w$ if $w > 0$. In fact, the heatmaps appear to become less noisy, as is depicted in Figure 5. However, noise is not fully suppressed. This is due to the fact, that the LRP γ-rule further suppresses contradicting contributions, because of the denominator in Equation (A.3).
In order to further improve visualization and to be comparable to LRP, the $\flat$-rule is also applied to the first layer. As can be seen in the heatmap of Figure 5, features become more visible and noise is also suppressed.

3.2.3 LIME

The application of LIME consists of four steps: (1) generation of a random interpretable state, (2) perturbation of the input image, (3) evaluation and weighting of the result, and (4) fitting

a surrogate model.

Random State Generation In order to find the relevance of superpixels, an interpretable data representation is needed. Given n superpixels, the interpretable state corresponds to a vector s of length n with entries either one or zero. Here, an entry of one corresponds to a superpixel being visible and unperturbed. A state vector $s \in \{0,1\}^n$, being aligned to the image superpixels, is generated by drawing n-times from a Bernoulli distribution with probability $p = 0.5$.
Regarding superpixel generation, the Simple Linear Iterative Clustering (SLIC) algorithm is used [Kim19] (see Figure 6 for an example). SLIC has been shown by Schallner et al. to be an effective algorithm for large and complex images using LIME [Sch+19]. In order to detect small objects in the heatmap, a resolution of $n = 250$ is chosen for the 512×512 pixel input.

Image Perturbation For real world images, it is hard to define a baseline image in general. A part of the image cannot be simply removed. The standard approach for LIME is setting the pixel values to the mean pixel value of the whole training data set (results in a gray color) [RSG16]. Other approaches consist of using the mean color of the superpixel, uniform noise, Gaussian blur or in-painting techniques.

Because edges contain a lot of information, an ideal perturbation would remove the edges. Color can contain also information, but by perturbing color, additional edges might be inserted. Thus, it is sensible to use an in-painting technique that preserves color rather than introduces new edges [ASN20]. However, because in-painting is a complex task, the run time is increased. Using gray pixels is a fast approach, and introduces on average less new edges than any other color, because it represents the mean color of the data set. In the following, perturbation using gray color is always chosen for LIME.

Figure 6: Example for SLIC generated superpixels used for LIME and SHAP heatmaps.

Weighting LIME is fitting an explainable (in general linear) model to a local neighborhood around a to-be explained sample of interest. In order to improve locality, the evaluations of the model are weighted regarding the amount of perturbation. Here, the distance metric is described by the cosine distance. Flattening the image tensors, the cosine distance is given by

$$d(\vec{x}, \vec{z}) = 1 - \frac{\vec{x} \cdot \vec{z}}{\|\vec{x}\|\|\vec{z}\|},$$

with unperturbed image vector $\vec{x}$ and perturbed vector $\vec{z}$.

The weight w of the perturbed sample is calculated by using a squared exponential kernel with kernel width k:

$$w = e^{-\frac{d^2}{k^2}}.$$

A smaller kernel value k will result in a stronger focus on local samples, while using fewer samples with a high distance d. In the following, the default kernel value $k = 1$ is chosen, such that $w \in [1/e, 1]$.

Fitting The fit function is chosen to be a linear function, where the weights are calculated solving a Ridge regression problem. In order to have enough data to receive a heatmap with

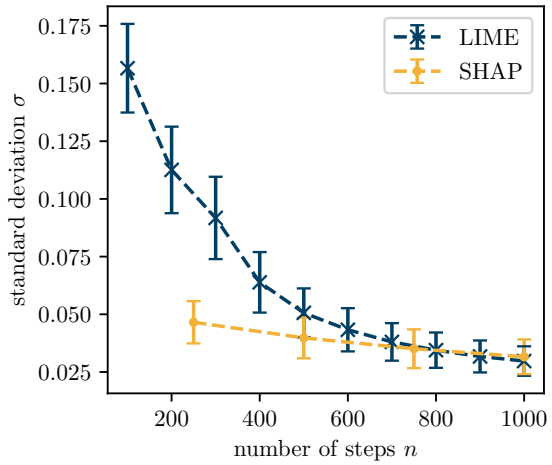

Figure 7: Standard deviation of heatmap superpixels (averaged over 10 samples) in relation to the number of steps n for LIME and SHAP. Error bars represent the standard error. The more steps (or forward passes) used in heatmap generation, the lower the standard deviation.

low variance, the number of samples is set to 1000. The decrease in variance of the heatmap for all superpixels (averaged over 10 samples) is displayed in Figure 7. It can be seen that the variance is slowly converging for a number of steps larger than 500. After determination of the coefficients of the linear function, the heatmap values are normalized to the interval $[-1, 1]$ by scaling.

3.2.4 SHAP

In order to resolve smaller objects in the heatmap and to be comparable to LIME, also $n = 250$ superpixels generated by the SLIC algorithm are used. Compared to fixed-size superpixels (e.g., squared superpixels), free-form SLIC superpixels have the advantage of being better aligned to the geometry of the input features. Thus, free-form superpixels are more informative and can better perturb both small and large features.

Further, the baseline is set to a gray image (mean color of the dataset) for the reasons discussed in Section 3.2.3. Regarding the number of steps, similar to LIME, SHAP is run four times corresponding to 1000 steps/forward passes. In this case, the heatmaps are deviating as much as for LIME (see Figure 7).

4 Comparison of methods

After adaption to the object detection task, the methods are in the following compared regarding human interpretability, faithfulness and applicability.

4.1 Human Interpretability

Defining or quantifying interpretability is challenging, because of its subjective nature [Sam+21]. Regarding an explanation, an expert expects more complex answers compared to a novice, who requires higher-level information. Given image data, the granularity of the heatmap represents the degree of detail and amount of information in the explanation. An explanation with a high granularity could depict specific features, like eyes or noses, compared to an explanation with low granularity showing coarser features, e.g., an entire head.

Measuring the information contained in an image can be done using the Shannon entropy e [GS85]:

$$e = -\sum_i p_i \log_2 p_i \tag{4.1}$$

with probability p_i of a pixel to have relevance value $r_i \in \mathcal{R}$. Here, $\mathcal{R}$ is the set containing all relevance values of an explanation.

method	single pixels	superpixels
LRP	11.2 ± 0.5	6.67 ± 0.25
DeepLIFT	12.4 ± 0.5	6.28 ± 0.21
LIME	7.81 ± 0.03	
SHAP	6.30 ± 0.21	

Table 1: Shannon entropy of heatmaps for all investigated XAI methods (averaged over 100 samples). LRP and DeepLIFT relevances are considered pixel-wise, as well as grouped into the SLIC superpixels used for LIME and SHAP.

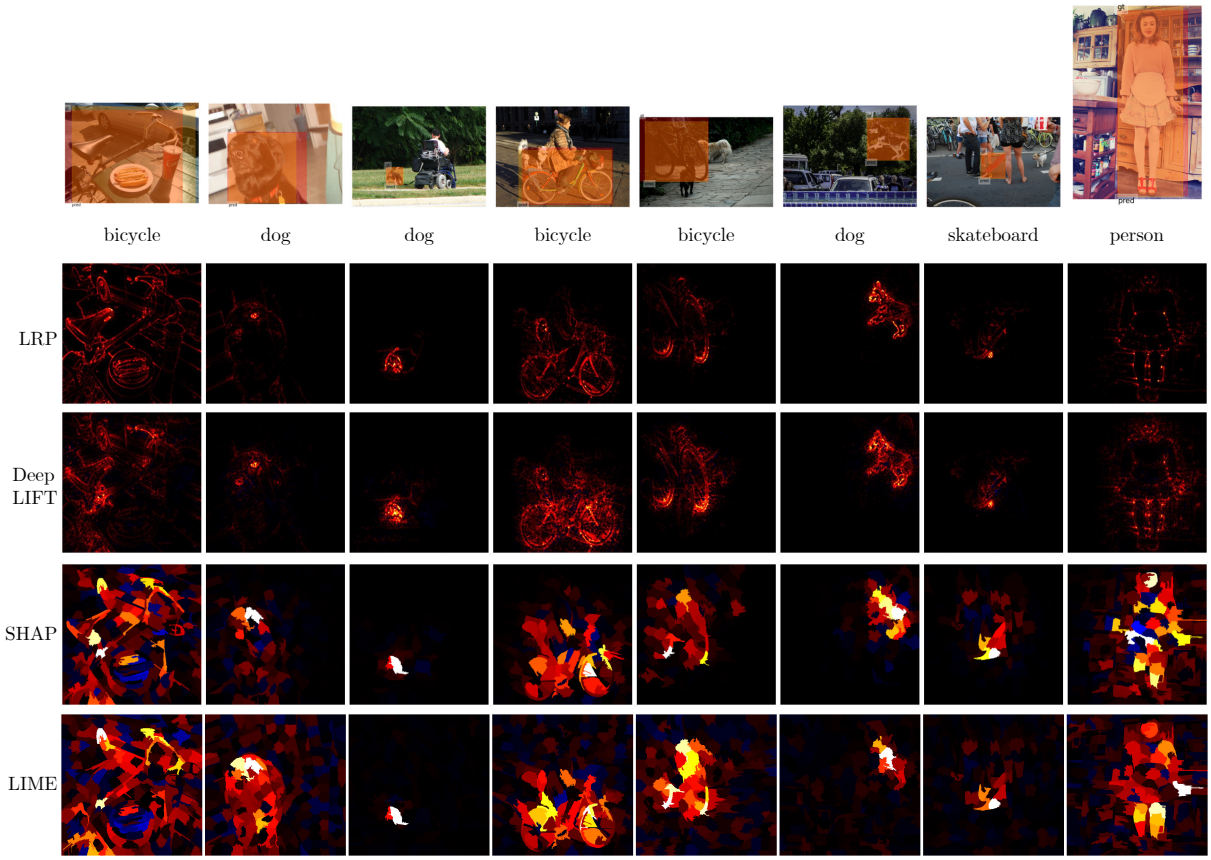

Figure 8: Explanations for eight bounding box class predictions (top row) with class name. Methods are LRP-$\gamma(0.1)$, DeepLIFT-$\gamma(0.1)$, SHAP and LIME from top to bottom.

As can be seen in Figure 8, LRP and DeepLIFT result in heatmaps with finer details compared to SHAP and LIME. This is also represented in entropy values of more than 11 for LRP and DeepLIFT, compared to LIME with about 7.8 and SHAP with 6.3 (see Table 1). Here, the slightly higher entropy value of DeepLIFT with 12.4 ± 0.5 compared to LRP with 11.2 ± 0.5 might be resulting from the visually more grainy heatmaps of DeepLIFT (see Figure 8).

The lower entropy value of SHAP compared to LIME follows from the difference in both algorithms. For SHAP, superpixels outside the receptive field of the output neuron do not contribute to the output, when individually added to the baseline image, and thus receive no relevance. However, for LIME, multiple superpixels are perturbed at the same time. Thus, also superpixels outside the receptive field receive relevance, as can be seen for the second image in Figure 8. Even if relevances are small, they count regarding the Shannon entropy (4.1). In fact, assuming that all 250 superpixels receive a different relevance value leads to an entropy of $e = -\log_2(\frac{1}{250}) \approx 7.97$, which is similar to the experimental LIME value of 7.81 ± 0.03. This effect also leads to a similar entropy for all LIME heatmaps, as is represented by a smaller standard error of 0.03 compared to SHAP with 0.21.

When LRP and DeepLIFT relevances are grouped into the superpixels used for LIME and SHAP, the entropy values are closer to SHAP with about 6.7 and 6.3, respectively. This is expected, as superpixels outside the receptive field of the output neuron also receive no relevance, because only contributing neurons are considered in the backward pass computation.

The fact that relevances, which are not perceptible by a human, influence the entropy shows, that the Shannon entropy is not an op-

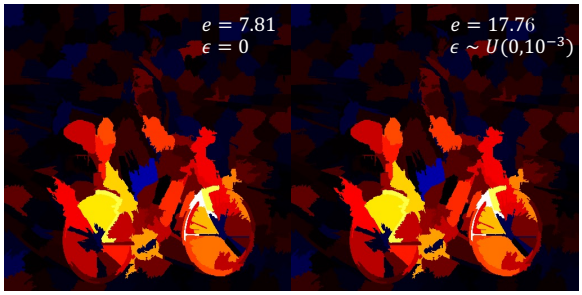

Figure 9: Entropy of a heatmap without noise (left) and with uniformly distributed noise $\varepsilon \sim U(0, 10^{-3})$ (right). Although being visually barely distinguishable, the heatmap's entropy e differs strongly.

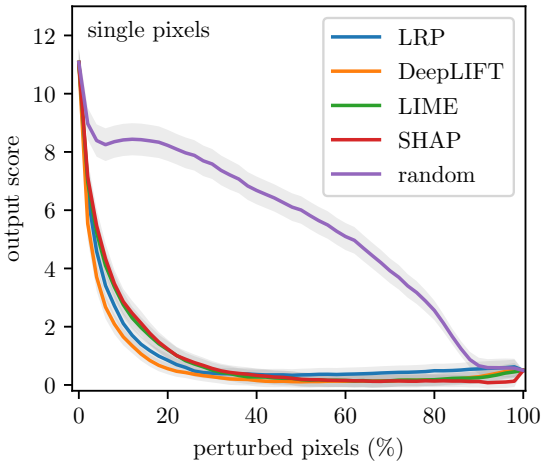

Figure 10: Pixel-flipping experiment: output (prediction) score over amount of relevant pixels set to gray for all XAI methods and random selection. LRP and DeepLIFT relevances are considered pixel-wise. The standard error resulting from 100 samples is depicted in light gray.

timal measure for human interpretability. This is visually shown in Figure 9, where a small amount of noise strongly influences Shannon entropy. A better human interpretability measure would be based on a human perceptual model or on cognitive experiments [Lag+19].

4.2 Faithfulness

Regarding faithfulness, it is important to investigate, whether relevant pixels of a heatmap are actually relevant for the prediction output. In order to check and compare faithfulness, a pixel-flipping and localization experiment is performed in the following.

4.2.1 Pixel-flipping

In the pixel-flipping experiment, starting with the most relevant pixels, all pixels are successively set to zero (gray color). Measuring the prediction (output) score, a faithful explanation is represented by a low area under curve (AUC). As depicted in Figure 10, all methods in the pixel-flipping experiment are similarly faithful, and outperform random selection. Here, the curves of the backpropagation-based methods are characterized by a slightly stronger decrease in the output score compared to the perturbation-based methods. This is expected, because LRP and DeepLIFT resolve small scale features better. Features are thus perturbed earlier than for SHAP and LIME.

Measuring the area under curve (AUC), DeepLIFT achieves the best value with 0.79 compared to 1.06 for the other methods. One reason for DeepLIFT to perform better than LRP (despite the similar heatmaps) consists in an increased granularity of the heatmaps (see Figure 8) leading to a more extensive perturbation of features.

In order to compare the methods on the same heatmap resolution, the relevances for DeepLIFT and LRP are also grouped into the superpixels used for LIME and SHAP. Performing the pixel-flipping experiment again, the backpropagation-based methods achieve now higher AUC values of 1.83 (LRP) and 1.80 (DeepLIFT), as can be seen in Figure 11 and Table 2. The reason for the lower AUC score might be two-fold.
On the one hand, while applying rules favoring positive contributions for LRP and DeepLIFT, negative contributions are slightly suppressed. When perturbing the missing negative relevant superpixels, the output score is thus increased.
Further, the perturbation-based methods are

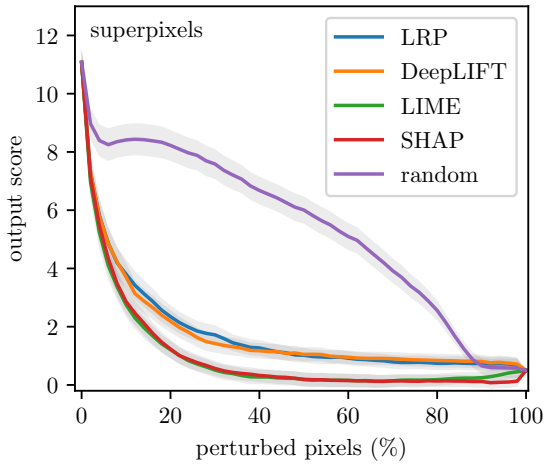

Figure 11: Pixel-flipping experiment: output (prediction) score over amount of relevant pixels set to gray for all XAI methods and random selection. LRP and DeepLIFT relevances are grouped into the SLIC superpixels used for LIME and SHAP. The standard error resulting from 100 samples is depicted in light gray.

optimized for the usage of superpixels and the procedure of the pixel-flipping experiment. In fact, SHAP and LIME work exactly by investigating how the perturbation of all features inside a superpixel directly influence the output score.

It is to note, that a reduction of noise in heatmaps is crucial for the pixel-flipping experiment. Noise leads to relevant features receiving both negative and positive scores, i.e., parts of a heatmap (being actually solely positively relevant) also constitute of negative relevance. As a result, a method's curve in Figure 10 is described by a high plateau that is only decreasing when the (falsely) negative relevant pixels are perturbed.

4.2.2 Localization

Inspired by other works [FV17; Zha+18], faithfulness can be investigated by the amount of relevance inside the bounding box of a classified object. The idea is, that the model pays attention mostly to the object it is predicting. This is assumed in the following, but is not always the case and depends on the model's functioning, as can be seen in Figure 1.

Aiming for a high amount of relevance inside bounding boxes, LRP and SHAP achieve the best results, with approximately 67 % of all positive relevance inside a bounding box (see Table 3). DeepLIFT and LIME attain lower scores with around 64 and 57 %, respectively. Here, LIME results with the lowest score, possibly because of the already discussed fact, that superpixels outside the receptive field receive relevance.
For DeepLIFT, the higher granularity of the heatmaps, compared to LRP, might result in pixels with relevance outside a bounding box. This can be seen in the fifth image of Figure 8, where relevant pixels lie outside the bounding box for class 'bicycle'.

method	single pixels	superpixels
LRP	1.06	1.83
DeepLIFT	0.79	1.80
LIME	1.06	
SHAP	1.06	
random	5.41	

Table 2: Area under curve (AUC) of the pixel-flipping experiment for all XAI methods and random selection. LRP and DeepLIFT relevances are considered pixel-wise, as well as grouped into the SLIC superpixels used for LIME and SHAP.

method	single pixels	superpixels
LRP	67.2 ± 2.2 %	62.3 ± 2.5 %
DeepLIFT	63.8 ± 2.5 %	59.9 ± 2.5 %
LIME	57.3 ± 2.5 %	
SHAP	66.9 ± 2.5 %	

Table 3: Amount of relevance within the bounding box of interest for all investigated XAI methods (averaged over 100 samples). LRP and DeepLIFT relevances are considered pixel-wise, as well as grouped into the SLIC superpixels used for LIME and SHAP.

In general, it is expected, that the backpropagation-based methods perform better than perturbation-based methods, because of the size of the superpixels. Large superpixels that cover relevant features might protrude beyond the borders of a bounding box. This effect can be seen when repeating the localization experiment with all relevances being aggregated into the SLIC superpixels. The scores for LRP and DeepLIFT drop around 5 and 4 %, respectively (see Table 3).

4.3 Applicability

Applicability can be a key criterion for the application of XAI. SHAP and LIME are popular in industry [LPK21], because they are model-agnostic, intuitive and easy to implement. DeepLIFT and LRP on the other hand require adaption to complex model architectures. This is also the case for the ℓ_2-Norm layer of the SSD model.

However, DeepLIFT and LRP use a modified backward pass to calculate the explanations and thus can also be used to extract relevances of intermediate neurons and weights. This can, for example, be used for pruning [Yeo+21] or quantization [SHT21].

Further, modified backward passes are an efficient and fast way to calculate relevances. Because of the input image size of 512×512 pixels, SHAP and LIME require a high number of superpixels and consequently many sample perturbations for stable heatmaps. In fact, the run time per explanation is more than two magnitudes larger for SHAP and LIME compared to DeepLIFT and LRP (see Table 4).

Even though LIME and SHAP use the same amount of perturbation samples, namely 1000, LIME requires about 50% more time compared to SHAP. This is, for example, caused by the sample weighting and the final linear fitting step. The run time evaluation is performed using an NVIDIA Titan V graphics card.

method	time per explanation
LRP	(110.5 ± 1.2) ms
DeepLIFT	(96.7 ± 0.9) ms
LIME	$(38.8 \pm 0.8) \cdot 10^3$ ms
SHAP	$(25.2 \pm 1.0) \cdot 10^3$ ms

Table 4: Mean run time for all investigated XAI methods (averaged over 100 samples). For better comparison, the Captum implementation of LRP is used.

5 Conclusion

Whereas LIME and SHAP are model-agnostic, DeepLIFT and LRP require to be adapted to the SSD architecture, i.e., the ℓ_2-Norm layer. However, for LIME and SHAP, several parameters such as the number of perturbation steps, perturbation technique or superpixel size have to be chosen. It still remains an open question how image data is ideally perturbed. Setting pixels to gray color is a fast and simple technique, but introduces artifacts and thus departs from the data manifold.

The perturbation-based XAI methods LIME and SHAP compute a prediction for each perturbation. In contrast, the backpropagation-based methods DeepLIFT and LRP require only one forward and backward pass. This leads to a more than two magnitudes lower run time per explanation. Run time can be a key factor, when hundreds of samples or whole datasets are to be explained.

The standard rules for LRP and DeepLIFT are not ideal for deep neural networks, such as SSD, because of gradient shattering leading to noisy heatmaps. Where LRP offers extended rules (e.g., γ-rule) to tackle the problem, DeepLIFT's implementation in Captum does not offer any such solution yet. In order to improve explanations, DeepLIFT has been modified, similar to the LRP γ-rule: Positive contributions are favored by scaling positively contributing weights. This modification leads to clearer heatmaps, however noise is not fully suppressed. The LRP γ-rule does not only favor stronger contributions, but also suppresses

contradicting outputs. This is not achieved by mere scaling of weights. DeepLIFT explanations can be further improved regarding noise and visibility using the LRP $\flat$-rule. All in all, the application of the LRP rules to DeepLIFT lead to a similar performance as LRP in all comparison tests. It is to note, that an implementation of the RevealCancel rule for DeepLIFT in Captum might also improve heatmaps.

To summarize, all methods have been shown to be faithful in the experiments, with perturbation-based methods being easier to apply for complex architectures than backpropagation-based methods. Here, appropriate hyperparameters and propagation rules are crucial. However, the backpropagation-based methods DeepLIFT and LRP are more efficient and heatmaps visualize individual features, as they are not limited by a superpixel size.

References

[16] *General Data Protection Regulation*. European Commission. Apr. 27, 2016. URL: https://gdpr-info.eu.

[Ade+18] Julius Adebayo et al. "Sanity Checks for Saliency Maps." In: *Advances in Neural Information Processing Systems* 31 (2018).

[And+21] Christopher J. Anders et al. "Software for Dataset-wide XAI: From Local Explanations to Global Insights with Zennit, CoRelAy, and ViRelAy." In: *CoRR* abs/2106.13200 (2021).

[Arr+20] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." In: *Information Fusion* 58 (2020), pp. 82–115.

[ASN20] Chirag Agarwal, Dan Schonfeld, and Anh Nguyen. "Removing input features via a generative model to explain their attributions to an image classifier's decisions." In: (2020).

[Bac+15] Sebastian Bach et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." In: *PloS one* 10.7 (2015), e0130140.

[Bac+16] Sebastian Bach et al. "Controlling explanatory heatmap resolution and semantics via decomposition depth." In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 2271–2275.

[Bal+17] David Balduzzi et al. "The shattered gradients problem: If resnets are the answer, then what is the question?" In: *International Conference on Machine Learning*. PMLR. 2017, pp. 342–350.

[CBR20] Zhi Chen, Yijie Bei, and Cynthia Rudin. "Concept whitening for interpretable image recognition." In: *Nature Machine Intelligence* 2.12 (2020), pp. 772–782.

[FV17] Ruth C Fong and Andrea Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3429–3437.

[GS85] SF Gull and J Skilling. "The entropy of an image." In: *Maximum-Entropy and Bayesian Methods in Inverse Problems*. Springer, 1985, pp. 287–301.

[Gui+18] Riccardo Guidotti et al. "A survey of methods for explaining black box models." In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[He+16] Kaiming He et al. "Deep residual learning for image recognition." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[HLA21] Tobias Huber, Benedikt Limmer, and Elisabeth André. "Benchmarking Perturbation-based Saliency Maps for Explaining Deep Reinforcement Learning Agents." In: *arXiv preprint arXiv:2101.07312* (2021).

[Hol+19] Andreas Holzinger et al. "Causability and explainability of artificial intelligence in medicine." In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.4 (2019), e1312.

[Kim19] Alchan Kim. *fast-slic*. https://github.com/Algy/fast-slic. 2019.

[Koh+20] Maximilian Kohlbrenner et al. "Towards best practice in explaining neural network decisions with LRP." In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–7.

[Lag+19] Isaac Lage et al. "An evaluation of the human-interpretability of explanation." In: *arXiv preprint arXiv:1902.00006* (2019).

[Lap+16] Sebastian Lapuschkin et al. "Analyzing classifiers: Fisher vectors and deep neural networks." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2912–2920.

[Lin+14] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context." In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[Liu+16] Wei Liu et al. “Ssd: Single shot multibox detector.” In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[LL17] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions.” In: *Proceedings of the 31st international conference on neural information processing systems*. 2017, pp. 4768–4777.

[LPK21] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. “Explainable ai: A review of machine learning interpretability methods.” In: *Entropy* 23.1 (2021), p. 18.

[Mon+19] Grégoire Montavon et al. “Layer-wise relevance propagation: an overview.” In: *Explainable AI: interpreting, explaining and visualizing deep learning* (2019), pp. 193–209.

[MSM18] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding deep neural networks.” In: *Digital Signal Processing* 73 (2018), pp. 1–15.

[N+19] Kokhlikyan N. et al. *Facebook Captum*. `https://github.com/pytorch/captum`. 2019.

[Pan15] Avneet Pannu. “Artificial intelligence and its application in different areas.” In: *Artificial Intelligence* 4.10 (2015), pp. 79–84.

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. “Rise: Randomized input sampling for explanation of black-box models.” In: *arXiv preprint arXiv:1806.07421* (2018).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “" Why should i trust you?" Explaining the predictions of any classifier.” In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[Sam+16] Wojciech Samek et al. “Evaluating the visualization of what a deep neural network has learned.” In: *IEEE transactions on neural networks and learning systems* 28.11 (2016), pp. 2660–2673.

[Sam+21] Wojciech Samek et al. “Explaining deep neural networks and beyond: A review of methods and applications.” In: *Proceedings of the IEEE* 109.3 (2021), pp. 247–278.

[Sch+19] Ludwig Schallner et al. “Effect of Superpixel Aggregation on Explanations in LIME–A Case Study with Biological Data.” In: *arXiv preprint arXiv:1910.07856* (2019).

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. “Learning important features through propagating activation differences.” In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3145–3153.

[SHT21] Muhammad Sabih, Frank Hannig, and Jürgen Teich. “Fault-Tolerant Low-Precision DNNs using Explainable AI.” In: *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE. 2021, pp. 166–174.

[SZ14] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” In: *arXiv preprint arXiv:1409.1556* (2014).

[TG20] Erico Tjoa and Cuntai Guan. “A survey on explainable artificial intelligence (xai): Toward medical xai.” In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[Yeo+21] Seul-Ki Yeom et al. “Pruning by explaining: A novel criterion for deep neural network pruning.” In: *Pattern Recognition* 115 (2021), p. 107899.

[Zha+18] Jianming Zhang et al. “Top-down neural attention by excitation backprop.” In: *International Journal of Computer Vision* 126.10 (2018), pp. 1084–1102.

A LRP-rules

All used LRP-rules are listed in the following.

ϵ-rule Using the ϵ-rule is close to propagation of the gradient through the network. Only neurons with a very small output are suppressed by the ϵ term in the denominator. Here, ϵ is of the same sign as the output $z_k = \sum_{0,j} a_j w_{jk}$ with activation a_j and weight w_{jk}. The ϵ-rule is given by

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,i} a_i w_{ik} + \epsilon} R_k. \tag{A.1}$$

γ-rule In a deep neural network with ReLU non-linearities, the γ-rule is used to suppress the noise in the heatmap:

$$R_j = \sum_k \frac{a_j \left(w_{jk}^- + (1+\gamma) w_{jk}^+ \right)}{\sum_{0,i} a_i \left(w_{ik}^- + (1+\gamma) w_{ik}^+ \right)} R_k \tag{A.2}$$

with $\gamma > 0$ in order to strengthen the positive contributions. Here, $^+$ and $^-$ denote all positive or negative values, respectively.

z^+-rule In a deep neural network, the z^+-rule is strongly suppressing noise in heatmaps by only following positively contributing activations throughout the network.

$$R_j = \sum_k \frac{\left(a_j w_{jk} \right)^+}{\sum_{0,i} (a_i w_{ik})^+} R_k \,. \tag{A.3}$$