Evaluating the Visualization of What a Deep Neural Network Has Learned

Wojciech Samek, *Member, IEEE*, Alexander Binder, *Member, IEEE*, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller, *Member, IEEE*

***Abstract*— Deep neural networks (DNNs) have demonstrated impressive performance in complex machine learning tasks such as image classification or speech recognition. However, due to their multilayer nonlinear structure, they are not transparent, i.e., it is hard to grasp *what* makes them arrive at a particular classification or recognition decision, given a new unseen data sample. Recently, several approaches have been proposed enabling one to understand and interpret the reasoning embodied in a DNN for a single test image. These methods quantify the "importance" of individual pixels with respect to the classification decision and allow a visualization in terms of a heatmap in pixel/input space. While the usefulness of heatmaps can be judged subjectively by a human, an objective quality measure is missing. In this paper, we present a general methodology based on region perturbation for evaluating ordered collections of pixels such as heatmaps. We compare heatmaps computed by three different methods on the SUN397, ILSVRC2012, and MIT Places data sets. Our main result is that the recently proposed layer-wise relevance propagation algorithm qualitatively and quantitatively provides a better explanation of what made a DNN arrive at a particular classification decision than the sensitivity-based approach or the deconvolution method. We provide theoretical arguments to explain this result and discuss its practical implications. Finally, we investigate the use of heatmaps for unsupervised assessment of the neural network performance.**

***Index Terms*— Convolutional neural networks, explaining classification, image classification, interpretable machine learning, relevance models.**

I. Introduction

DEEP neural networks (DNNs) are powerful methods for solving large-scale real-world problems such as automated image classification [1]–[4], natural language processing [5], [6], human action recognition [7], [8], or physics [9] (see also [10]). Since DNN training methodologies (unsupervised pretraining, dropout, parallelization, GPUs, etc.) have been improved [11], DNNs are recently able to harvest extremely large amounts of training data and can thus achieve record performances in many research fields. At the same time, DNNs are generally conceived as black box methods, and users might consider this lack of transparency a drawback in practice. Namely, it is difficult to intuitively and quantitatively understand the result of DNN inference, i.e., for an *individual* novel input data point, *what* made the trained DNN model arrive at a particular response. Note that this aspect differs from feature selection [12], where the question is: which features are on average salient for the *ensemble* of training data?

Only recently, the transparency problem has been receiving more attention for general nonlinear estimators [13]–[16]. Several methods have been developed to understand what a DNN has learned [17], [18]. In DNN a large body of work is dedicated to visualize particular neurons or neuron layers [1], [19]–[24], We focus here on methods that visualize the impact of particular regions of a given and fixed single image for a prediction of this image. Zeiler and Fergus [19] have proposed in their work a network propagation technique to identify patterns in a given input image that are linked to a particular DNN prediction. This method runs a backward algorithm that reuses the weights at each layer to propagate the prediction from the output down to the input layer, leading to the creation of meaningful patterns in input space. This approach was designed for a particular type of neural network, namely, convolutional nets with max pooling and rectified linear units (ReLUs). A limitation of the deconvolution method is the absence of a particular theoretical criterion that would directly connect the predicted output to the produced pattern in a quantifiable way. Furthermore, the usage of image-specific information for generating the backprojections in this method is limited to max-pooling layers alone. Further previous work has focused on understanding nonlinear learning methods such as DNNs or kernel methods [14], [25], [26] essentially by sensitivity analysis in the sense of scores based on partial derivatives at the given sample. Partial derivatives look at local sensitivities detached from the decision boundary of the classifier. Simonyan *et al*. [26] applied partial derivatives for visualizing input sensitivities in images classified by a DNN. Note that although [26] describes a Taylor series, it relies on partial derivatives at the given image for the computation of results. In a strict sense, partial derivatives do not explain a classifier's decision (*"what speaks for the presence of a*

Manuscript received June 1, 2016; revised August 7, 2016; accepted August 9, 2016. Date of publication August 25, 2016; date of current version October 16, 2017. This work was supported in part by the Brain Korea 21 Plus Program through the National Research Foundation of Korea Funded by the Ministry of Education, in part by DFG under Grant MU 987/17-1, and in part by the German Ministry for Education and Research as Berlin Big Data Center under Grant 01IS14013A. *(Wojciech Samek and Alexander Binder contributed equally to this work.) (Corresponding authors: Wojciech Samek; Alexander Binder; Klaus-Robert Müller.)*

W. Samek and S. Lapuschkin are with the Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany (e-mail: wojciech.samek@hhi.fraunhofer.de; sebastian.lapuschkin@hhi.fraunhofer.de).

A. Binder is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore 487372, and also with the Technische Universität Berlin, 10587 Berlin, Germany (e-mail: alexander_binder@sutd.edu.sg).

G. Montavon is with the Technische Universität Berlin, 10587 Berlin, Germany (e-mail: gregoire.montavon@tu-berlin.de).

K.-R. Müller is with the Technische Universität Berlin, 10587 Berlin, Germany, and also with the Department of Brain and Cognitive Engineering, Korea University, Seoul 136-713, South Korea (e-mail: klaus-robert.mueller@tu-berlin.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2599820

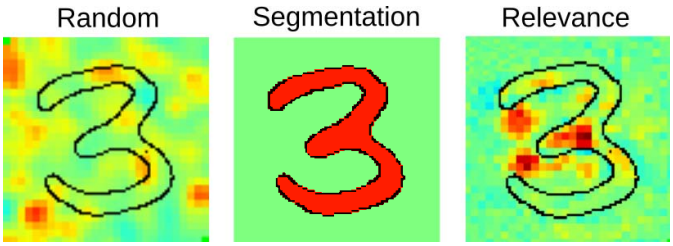


Fig. 1. Comparison of three exemplary heatmaps for the image of a "3." Left: the randomly generated heatmap lacks interpretable information. Middle: the segmentation heatmap (binary values) focuses on the whole digit without indicating what parts of the image were particularly relevant for classification. Since it does not suffice to consider only the highlighted pixels for distinguishing an image of a "3" from images of an "8" or a "9" this heatmap is not useful for explaining classification decisions. Right: a relevance heatmap indicates which parts of the image are used by the classifier. Here the heatmap reflects human intuition very well because the horizontal bar together with the missing stroke on the left are strong evidence that the image depicts a "3" and not any other digit.

car in the image"), but rather tell us *what change would make the image more or less belong to the category car*. As shown later, these two types of explanations lead to very different results in practice. An approach, layer-wise relevance propagation (LRP), which is applicable to arbitrary types of neural unit activities (even if they are noncontinuous) and to general DNN architectures (and Fisher vector classifiers [27]) has been proposed in [28]. This paper aims at explaining the difference between a prediction $f(x)$ relative to the neutral state $f(x) = 0$. The LRP method relies on a conservation principle to propagate the prediction back without using gradients. This principle ensures that the network output activity is fully redistributed through the layers of a DNN onto the input variables, i.e., neither positive nor negative evidence is lost.

In the following, we will denote the visualizations produced by the above methods as heatmaps. While per se a heatmap is an interesting and intuitive tool that can already allow achieving transparency, it is difficult to quantitatively evaluate the quality of a heatmap. In other words we may ask: what exactly makes a "good" heatmap? A human may be able to intuitively assess the quality of a heatmap, e.g., by matching with a prior of what is regarded as being relevant (see Fig. 1). For practical applications, however, an automated objective and quantitative measure for assessing heatmap quality becomes necessary. Note that the validation of heatmap quality is important if we want to use it as input for further analysis. For example, we could run computationally more expensive algorithms only on relevant regions in the image, where relevance is detected by a heatmap.

In this paper, we contribute by the following.

1) Pointing to the issue of how to objectively evaluate the quality of heatmaps. To the best of our knowledge this question has not been raised so far.
2) Introducing a generic framework for evaluating heatmaps, which extends the approach in [28] from binary inputs to color images.
3) Comparing three different heatmap computation methods on three large data sets and noting that the relevance-based LRP algorithm [28] is more suitable for explaining the classification decisions of DNNs than the sensitivity-based approach [26] and the deconvolution method [19].
4) Investigating the use of heatmaps for assessment of the neural network performance.

The following section briefly introduces three existing methods for computing heatmaps. Section III discusses the heatmap evaluation problem and presents a generic framework for this task. Two experimental results are presented in Section IV. The first experiment compares different heatmapping algorithms on SUN397 [29], ILSVRC2012 [30], and MIT Places [31] data sets and the second experiment investigates the correlation between heatmap quality and neural network performance on the CIFAR-10 data set [32]. We conclude this paper in Section V and give an outlook.

II. Understanding DNN Prediction

In the following, we focus on images, but the presented techniques are applicable to any type of input domain whose elements can be processed by a neural network.

Let us consider an image $\boldsymbol{x} \in \mathbb{R}^d$, decomposable as a set of pixel values $\boldsymbol{x} = \{x_p\}$ where p denotes a particular pixel, and a classification function $f : \mathbb{R}^d \to \mathbb{R}^+$. The function value $f(\boldsymbol{x})$ can be interpreted as a score indicating the certainty of the presence of a certain type of object(s) in the image. Such functions can be learned very well by a DNN. Throughout this paper we assume neural networks to consist of multiple layers of neurons, where neurons are activated as

$$a_j^{(l+1)} = \sigma\left(\sum_i z_{ij} + b_j^{(l+1)}\right) \tag{1}$$

with

$$z_{ij} = a_i^{(l)} w_{ij}^{(l,l+1)}. \tag{2}$$

The sum operator runs over all the lower layer neurons that are connected to neuron j, where $a_i^{(l)}$ is the activation of a neuron i in the previous layer, and where z_{ij} is the contribution of neuron i at layer l to the activation of the neuron j at layer $l+1$. The function σ is a nonlinear monotonously increasing activation function, $w_{ij}^{(l,l+1)}$ is the weight, and $b_j^{(l+1)}$ is the bias term.

A *heatmap* $\boldsymbol{h} = \{h_p\}$ assigns each pixel p a value $h_p = \mathcal{H}(\boldsymbol{x}, f, p)$ according to some function $\mathcal{H}$, typically derived from a class discriminant f. Since $\boldsymbol{h}$ has the same dimensionality as $\boldsymbol{x}$, it can be visualized as an image. In the following we review three recent methods for computing heatmaps, all of them performing a backward propagation pass on the network: 1) a sensitivity analysis based on neural network partial derivatives; 2) the so-called deconvolution method; and 3) the LRP algorithm. Fig. 2 briefly summarizes the methods.

A. Sensitivity Heatmaps

A well-known tool for interpreting nonlinear classifiers is sensitivity analysis [14]. It was used in [26] to compute saliency maps of images classified by neural networks. In this approach, the sensitivity of a pixel h_p is computed by using

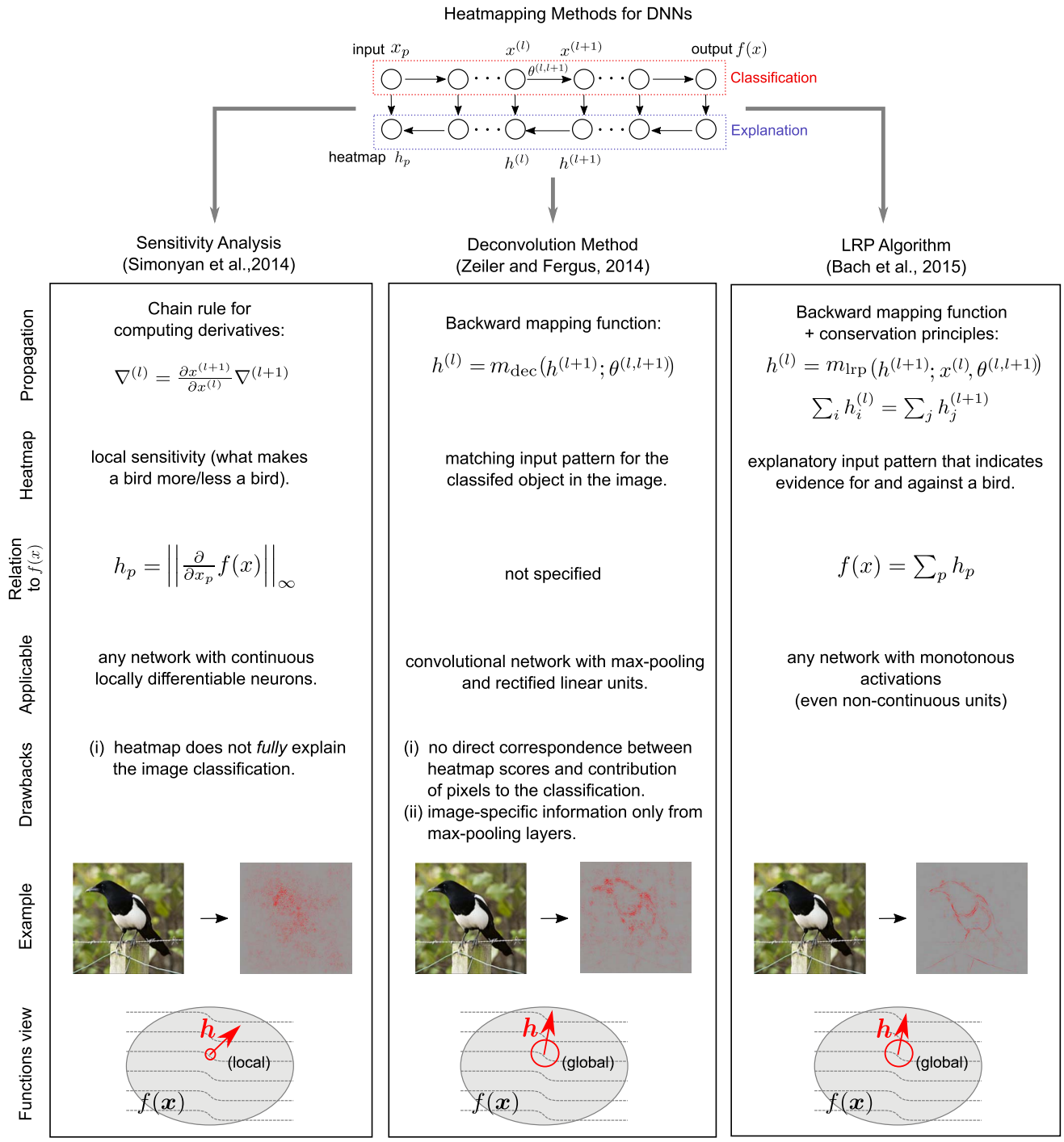


Fig. 2. Comparison of the three heatmap computation methods used in this paper. Left: sensitivity heatmaps are based on partial derivatives, i.e., measure which pixels, when changed, would make the image belong less or more to a category (local explanations). The method is applicable to generic architectures with differentiable units. Middle: the deconvolution method applies a convolutional network g to the output of another convolutional network f. Network g is constructed in a way to "undo" the operations performed by f. Since negative evidence is discarded and scores are not normalized during the backpropagation, the relation between heatmap scores and the classification output $f(x)$ is unclear. Right: LRP exactly decomposes the classification output $f(x)$ into pixel relevances by observing the layer-wise conservation principle, i.e., evidence for or against a category is not lost. The algorithm does not use gradients and is therefore applicable to generic architectures (including nets with noncontinuous units). LRP globally explains the classification decision and heatmap scores have a clear interpretation as evidence for or against a category.

the norm $\|\cdot\|_{\ell_q}$ over partial derivatives ([26] used $q = \infty$) for the color channel c of a pixel p

$$h_p = \left\| \left(\frac{\partial}{\partial x_{p,c}} f(\boldsymbol{x}) \right)_{c \in (r,g,b)} \right\|_{\ell_q}. \tag{3}$$

This quantity measures how much small changes in the pixel value locally affect the network output. Large values of h_p denote pixels that largely affect the classification function f if changed. Note that the direction of change (i.e., sign of the partial derivative) is lost when using the norm. Partial

derivatives are obtained efficiently by running the backpropagation algorithm [33] throughout the multiple layers of the network. The backpropagation rule from one layer to another layer, where $x^{(l)}$ and $x^{(l+1)}$ denote the neuron activities at two consecutive layers, is given by

$$\frac{\partial f}{\partial x^{(l)}} = \frac{\partial x^{(l+1)}}{\partial x^{(l)}} \frac{\partial f}{\partial x^{(l+1)}}. \tag{4}$$

The backpropagation algorithm performs the following operations in the various layers.

1) *Unpooling:* The gradient signal is redirected onto the input neuron(s) to which the corresponding output neuron is sensitive. In the case of max pooling, the input neuron in question is the one with the maximum activation value.
2) *Nonlinearity:* Denoting by $z_i^{(l)}$ the preactivation of the ith neuron of the lth layer, backpropagating the signal through a ReLU defined by the map $z_i^{(l)} \to \max(0, z_i^{(l)})$ corresponds to multiplying the backpropagated gradient signal by the step function $1_{\{z_i^{(l)}>0\}}$. The multiplication of the signal by the step function makes the backward mapping discontinuous, and consequently strongly local.
3) *Filtering:* The gradient signal is convolved by a transposed version of the convolutional filter used in the forward pass.

In the experiments, we compute heatmaps using (3) with the norms $q = \{2, \infty\}$.

B. Deconvolution Heatmaps

Another method for heatmap computation was proposed in [19] and uses a process termed deconvolution. Similar to the backpropagation method to compute the function's gradient, the idea of the deconvolution approach is to map the activations from the network's output back to pixel space using a backpropagation rule

$$R^{(l)} = m_{\text{dec}}(R^{(l+1)}; \theta^{(l,l+1)}). \tag{5}$$

Here, $R^{(l)}$, $R^{(l+1)}$ denote the backward signal as it is backpropagated from one layer to the previous layer, m_{dec} is a predefined function that may be different for each layer, and $\theta^{(l,l+1)}$ is the set of parameters connecting two layers of neurons. This method was designed for a convolutional net with max pooling and ReLUs, but it could also be adapted in principle for other types of architectures. The following set of rules is applied to compute deconvolution heatmaps.

1) *Unpooling:* The locations of the maxima within each pooling region are recorded and these recordings are used to place the relevance signal from the layer above into the appropriate locations. For deconvolution, this seems to be the only place besides the classifier output where image information from the forward pass is used, in order to arrive at an image-specific explanation.
2) *Nonlinearity:* The relevance signal at a ReLU layer is passed through a ReLU function during the deconvolution process.
3) *Filtering:* In a convolution layer, the transposed versions of the trained filters are used to backpropagate the relevance signal. This projection does not depend on the neuron activations $x^{(l)}$.

The unpooling and filtering rules are the same as those derived from gradient propagation (i.e., those used in Section II-A). The propagation rule for the ReLU nonlinearity differs from backpropagation: here, the backpropagated signal is not multiplied by a discontinuous step function, but is instead passed through a rectification function similar to the one used in the forward pass. Note that unlike the indicator function, the rectification function is continuous. For deconvolution, we apply the same color channel pooling methods (2-norm, ∞-norm) as for sensitivity analysis.

C. Relevance Heatmaps

LRP [28] is a principled approach to decompose a classification decision into pixel-wise *relevances* indicating the contributions of a pixel to the overall classification score. The approach is derived from a layer-wise conservation principle [28], which forces the propagated quantity (e.g., evidence for a predicted class) to be preserved between neurons of two adjacent layers. Denoting by $R_i^{(l)}$ the relevance associated with the ith neuron of layer l and by $R_j^{(l+1)}$ the relevance associated with the jth neuron in the next layer, the conservation principle requires that

$$\sum_i R_i^{(l)} = \sum_j R_j^{(l+1)} \tag{6}$$

where the sums run over all the neurons of the respective layers. Applying this rule repeatedly for all the layers, the heatmap resulting from LRP satisfies $\sum_p h_p = f(\boldsymbol{x})$ where $h_p = R_p^{(1)}$ and is said to be consistent with the evidence for the predicted class. Stricter definitions of conservation that involve only subsets of neurons can further impose that relevance is locally redistributed in the lower layers. The propagation rules for each type of layer are given by the following.

1) *Unpooling:* Like for the previous approaches, the backward signal is redirected proportionally onto the location for which the activation was recorded in the forward pass.
2) *Nonlinearity:* The backward signal is simply propagated onto the lower layer, ignoring the rectification operation. Note that this propagation rule satisfies (6).
3) *Filtering:* Bach *et al.* [28] proposed two relevance propagation rules for this layer that satisfy (6). Let $z_{ij} = a_i^{(l)} w_{ij}^{(l,l+1)}$ be the weighted activation of neuron i onto neuron j in the next layer. The first rule is given by

$$R_i^{(l)} = \sum_j \frac{z_{ij}}{\sum_{i'} z_{i'j} + \epsilon \operatorname{sign}\left(\sum_{i'} z_{i'j}\right)} R_j^{(l+1)}. \tag{7}$$

The intuition behind this rule is that the lower layer neurons that mostly contribute to the activation of the higher layer neuron receive a larger share of the relevance R_j of the neuron j. The neuron i then collects the relevance associated with its contribution from all the upper layer

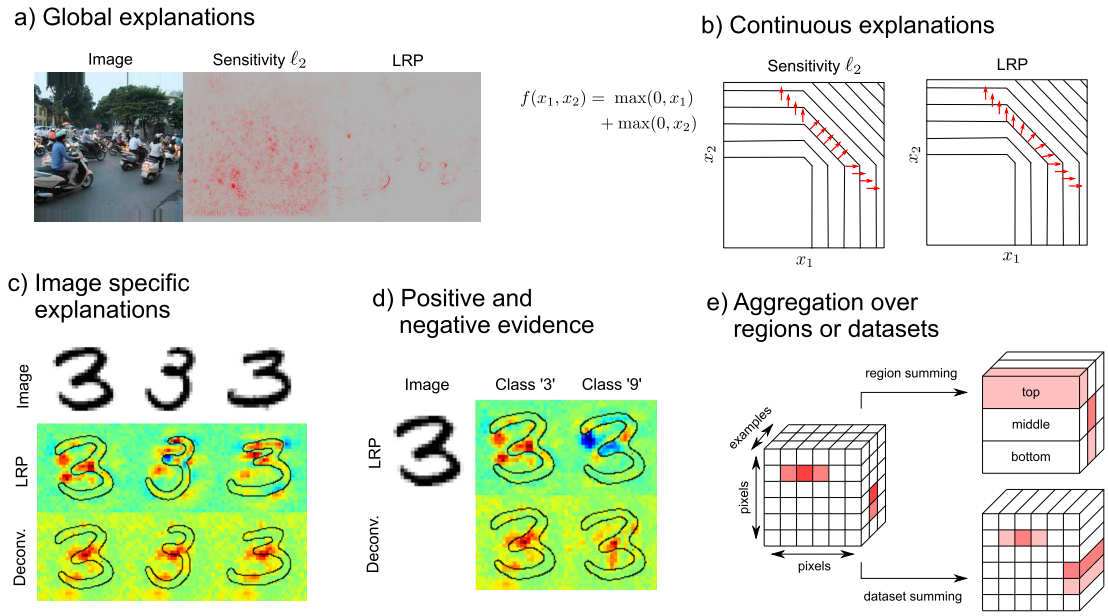


Fig. 3. Desirable properties of heatmapping methods. (a) LRP provides global explanations, i.e., indicates the features that explain the prediction "scooter," whereas sensitivity analysis provides local explanations, i.e., shows what would make the image less or more belong to the category "scooter." (b) LRP provides continuous explanations (red arrows) that change slowly with the input, whereas the explanations provided by sensitive analysis change abruptly due to discontinuities. (c) LRP provides image-specific explanations, because it takes into account the weights and the activations, whereas the deconvolution method only considers the weights and thus produces the same heatmaps for different input images. (d) LRP distinguishes between positive evidence supporting a prediction (red region) and negative evidence speaking against it (blue region), whereas deconvolution does not provide signed explanations. (e) The conservation property of LRP provided a meaningful normalization of the heatmaps, and thus allows one to aggregate the explanations over regions or data sets.

neurons j. A downside of this propagation rule (at least if $\epsilon = 0$) is that the denominator may tend to zero if the lower level contributions to neuron j cancel each other out. The numerical instability can be overcome by setting $\epsilon > 0$. However, in that case, the conservation idea is relaxated in order to gain better numerical properties. A way to achieve exact conservation is by separating the positive and negative activations in the relevance propagation formula, which yields the second formula

$$R_i^{(l)} = \sum_j \left(\alpha \cdot \frac{z_{ij}^+}{\sum_{i'} z_{i'j}^+} + \beta \cdot \frac{z_{ij}^-}{\sum_{i'} z_{i'j}^-} \right) R_j^{(l+1)}. \quad (8)$$

Here, z_{ij}^+ and z_{ij}^- denote the positive and negative parts of z_{ij}, respectively, such that $z_{ij}^+ + z_{ij}^- = z_{ij}$. We enforce $\alpha + \beta = 1$ in order for the relevance propagation equations to be conservative layer-wise. It should be emphasized that unlike gradient-based techniques, the LRP formula is applicable to nondifferentiable neuron activation functions. The LRP algorithm does not multiply its backward signal by a discontinuous function. Therefore, relevance heatmaps also favor the emergence of global features that allow for a full explanation of the class to be predicted.

In the experiments section, we use for consistency the same settings as in [28] without having optimized the parameters, namely, the LRP variant from (8) with $\alpha = 2$ and $\beta = -1$ (which will be denoted as LRP in the subsequent figures), and twice LRP from (7) with $\epsilon = 0.01$ and 100. An implementation of LRP is described in [34] and can be downloaded from `http://heatmapping.org`.

D. Theoretical Analysis

In the following we investigate the advantages and limitations of the presented heatmapping methods. We show that LRP has six desirable properties, which are not or only partly satisfied by sensitivity analysis and the deconvolution method.

1) Global Explanations: A desired property of heatmapping methods is that they provide global explanations, e.g., indicate the features that compose a given car. This property is satisfied by LRP and to some extent by the deconvolution method, but not by sensitivity analysis. The latter gives for every pixel a direction in RGB space in which the prediction increases or decreases, but it does not indicate directly whether a particular region contains evidence for or against the prediction made by a classifier. Thus, it provides *local* explanations, e.g., indicates what makes a given car look more/less like a car. Fig. 3(a) shows the qualitative difference between gradient- and LRP-type explanations. In terms of gradients, it is a valid explanation to put high norms on the empty street, because there exists a direction in the input space in which the classifier prediction can be increased by putting motor-bike like structures in there. However, from a global explanation perspective the streets are not very indicative of the class scooter in this particular image. This example shows that regions consisting of a pure background may have a notable sensitivity, which makes gradient-type explanations noisier than LRP and deconvolution heatmaps (see also Fig. 6).

2) Continuous Explanations: Another desired property of heatmapping methods is that they provide continuous explanations, i.e., small variations in the input should not result in large changes in the heatmap. Also, this property is not satisfied by gradient-type methods. The multiplication of the signal by an indicator function in the rectification layer (see Section II-A) makes the backward mapping of gradient-type methods discontinuous, i.e., they may abruptly switch from considering one feature as being highly relevant to considering another feature as being the most important one. Fig. 3(b) shows that sensitivity analysis of a 2-D function $f(x_1, x_2)$ results in discontinuous explanations (red arrows abruptly change direction), whereas LRP (and also deconvolution) does not show this behavior and thus provides more reliable explanations.

3) Image-Specific Explanations: Salient features represent average explanations of what distinguishes one image category from another. For individual images these explanations may be meaningless or even wrong. The deconvolution method implicitly takes into account only properties of individual images through the unpooling operation. The backprojection over the filtering layers is independent of the individual image. Thus, when applied to neural networks without a pooling layer, both the methods will not provide individual (image specific) explanations, but rather average salient features. LRP's rule for filtering layers on the other hand takes into account *both* the filter weights and the lower layer neuron activations. This allows for individual explanations even in a neural network without pooling layers. Fig. 3(c) demonstrates this property on a simple example. We compare the explanations provided by the deconvolution method and LRP for a neural network without pooling layers trained on the MNIST data set. One can see that LRP provides individual explanations for all the images in the sense that when the digit in the image is slightly rotated, then the heatmap adapts to this rotation and highlights the relevant regions of this particular rotated digit. The deconvolution heatmap on the other hand is not image specific because it only depends on the weights and not on the neuron activations. If pooling layers were present in the network, then the deconvolution approach would implicitly adapt to the specific image through the unpooling operation. Still we consider this information important to be included when backprojecting over filtering layers, because neurons with large activations for a specific image should be regarded as more relevant, and thus should backproject a larger share of the relevance.

4) Positive and Negative Evidence: In contrast to sensitivity analysis and the deconvolution method, LRP provides signed explanations and thus distinguishes between positive evidence, supporting the classification decision, and negative evidence, speaking against the prediction. Fig. 3(d) shows that LRP responses can be well interpreted in this way; the red and blue regions represent the positive and negative evidence, respectively. In particular, when backpropagating the (artificial) classification decision that the image has been classified as "9," LRP provides a very intuitive explanation, namely, that in the left upper part of the image the missing stroke closing the loop (blue region) speaks against the fact that this is a "9" whereas the missing stroke in the left lower part of the image (red region) supports this decision. The deconvolution method (and the gradient-type explanation) does not allow such interpretation.

5) Aggregating Over Regions or Data Sets: Aggregating explanations over image regions or over different data sets [see Fig. 3(e)] is a desired property requiring meaningful normalization of the pixel-wise scores. The explanations computed by sensitivity analysis and the deconvolution method are not normalized, so that aggregation may lead to meaningless results (e.g., when the heatmap of one image has values which are an order of magnitude larger than the values of other heatmaps). The LRP scores on the other hand are directly related to the classification output through the conservation principle, and thus are meaningfully normalized. This allows for a meaningful aggregation.

6) Relation to Classification Output: Another desirable property of heatmapping methods is an explicit mathematical relation between heatmap and the classification output, because this allows for an interpretation of the obtained scores. Such an explicit relation exists for the gradient-type approach and for the LRP method (see formula in Fig. 2). For the deconvolution method these relationships cannot be expressed analytically, because negative evidence ($R^{(l+1)} < 0$) is discarded during the backpropagation due to the application of the ReLU function and the backward signal is not normalized layer-wise, so that few dominant $R^{(l)}$ may largely determine the final heatmap scores.

Finally, an additional justification for the way LRP technically operates can be found in [35], where the method is shown for certain choices of parameters to perform a "deep Taylor decomposition" of the neural network function.

III. Evaluating Heatmaps

In this section, we introduce a set of methods to evaluate empirically the quality of a heatmapping technique. Although humans are able to intuitively assess the quality of a heatmap by matching with prior knowledge and experience of what is regarded as being relevant, defining objective criteria for heatmap quality is very difficult. In this paper, we refrain from mimicking the complex human heatmap evaluation process that includes attention, interest point models, and perception models of saliency [36]–[39] for the reason that we are interested in finding those regions that are relevant for a given classifier.

Rather than representing human reasoning, or matching some ground truth on what is important in an image, heatmaps should reflect the machine learning classifier's own "view" on the classification problem, more precisely, identify the pixels used by the classifier to support its decision. Thus, the heatmap quality does not only depend on the algorithms used to compute a heatmap, but also on the performance of the classifier, whose efficiency largely depends on the model being used, and the amount and quality of available training data. For example, if the training data does not contain images of the digit "3," then the classifier cannot know that the absence of strokes in the left part of the image (see example in Fig. 1) is important for distinguishing the digit "3" from

digits "8" and "9." Thus, explanations can only be as good as the data provided to the classifier. A random classifier will lead to uninformative heatmaps.

Note also that a heatmap differs from a segmentation mask (see Fig. 1) in several ways.

1) A heatmap can rightfully associate relevance with pixels outside the object to detect, for example, when the context of the object (e.g., water texture behind a boat) provides some useful information for classification.
2) Segmentation masks are binary (i.e., they essentially determine whether a pixel is part or not of the object to detect).

A heatmap, on the other hand, provides a gradation of pixel scores that correspond to the degree of importance of each pixel for determining the predicted class membership. Points of interest (e.g., eyes or small objects) might concentrate as much information for determining the class as larger surfaces. A heatmap can also associate negative values with pixels that contradict the prediction.

A. Heatmap Evaluation Framework

More formally, a heatmap is an array of pixel-wise scores $(R_p)_p$ that indicate which pixels are *relevant* for a classification decision [i.e., which pixels make $f(\boldsymbol{x})$ large]. A heatmap can be viewed as defining a subspace composed of pixels with high relevance scores, on which the function $f(\boldsymbol{x})$ must be scrutinized. We expect the pixels to which most relevance is associated to be those that are the most likely to destroy the value $f(\boldsymbol{x})$ if they are perturbed; in other words, we would like to test how fast the function value drops when moving in this subspace.

To test this expected behavior, we consider a greedy iterative procedure that consists of measuring how the class encoded in the image (e.g., as measured by the function f) disappears when we progressively remove information from the image $\boldsymbol{x}$, a process referred to as *region perturbation*, at the specified locations. The method is a generalization of the approach presented in [28], where the perturbation process is a state flip of the associated binary pixel values (single pixel perturbation). The method that we propose here applies more generally to *any* set of locations (e.g., local windows) and any local perturbation process such as local randomization or blurring.

We define a heatmap as an ordered set of locations in the image, where these locations might lie on a predefined grid

$$\mathcal{O} = (\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_L). \tag{9}$$

Each location $\boldsymbol{r}_p$ is for example a 2-D vector encoding the horizontal and vertical positions on a grid of pixels. The ordering can either be chosen at hand, or be induced by a heatmapping function $h_p = \mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_p)$, typically derived from a class discriminant f (see methods in Section II). The scores $\{h_p\}$ indicate how important the given location $\boldsymbol{r}_p$ of the image is for representing the image class. The ordering induced by the heatmapping function is such that for all indices of the ordered sequence $\mathcal{O}$, the following property holds:

$$(i < j) \Leftrightarrow (\mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_i) > \mathcal{H}(\boldsymbol{x}, f, \boldsymbol{r}_j)). \tag{10}$$

Thus, locations in the image that are most relevant for the class encoded by the classifier function f will be found at the beginning of the sequence $\mathcal{O}$. Conversely, regions of the image that are mostly irrelevant will be positioned at the end of the sequence.

We consider a region perturbation process that follows the ordered sequence of locations. We call this process *most relevant first* (MoRF). The recursive formula is

$$\begin{aligned} \boldsymbol{x}^{(0)}_{\text{MoRF}} &= \boldsymbol{x} \\ \forall\ 1 \le k \le L:\ \boldsymbol{x}^{(k)}_{\text{MoRF}} &= g\big(\boldsymbol{x}^{(k-1)}_{\text{MoRF}}, \boldsymbol{r}_k\big) \end{aligned} \tag{11}$$

where the function g removes information about the image $\boldsymbol{x}^{(k-1)}_{\text{MoRF}}$ at a specified location $\boldsymbol{r}_k$ (i.e., a single pixel or a local neighborhood) in the image. Throughout this paper, we use a function g that replaces all the pixels in an $m \times m$ neighborhood around $\boldsymbol{r}_k$ by randomly sampled (from uniform distribution) values. The choice of a uniform distribution follows our intention to use a model-free method to generate perturbations. A model-based estimation of probabilities for patches may result in biases due to the model assumptions. Using the uniform distribution ensures that we treat all the regions equally and explore the whole imaging space. Furthermore, it ensures that we evaluate the behavior of the classifier under perturbations that are off the data manifold.

When comparing different heatmaps using a fixed $g(\boldsymbol{x}, \boldsymbol{r}_k)$, our focus is typically only on the highly relevant regions (i.e., the sorting of the h_p values on the nonrelevant regions is not important). The quantity of interest in this case is the area over the MoRF perturbation curve (AOPC)

$$\text{AOPC} = \frac{1}{L+1}\left\langle \sum_{k=0}^{L} f\big(\boldsymbol{x}^{(0)}_{\text{MoRF}}\big) - f\big(\boldsymbol{x}^{(k)}_{\text{MoRF}}\big) \right\rangle_{p(\boldsymbol{x})} \tag{12}$$

where $\langle\cdot\rangle_{p(\boldsymbol{x})}$ denotes the average over all the images in the data set. An ordering of regions such that the most sensitive regions are ranked first implies a steep decrease of the graph of MoRF, and thus a larger AOPC.

IV. Experimental Results

In this section, we use the proposed heatmap evaluation procedure to compare heatmaps computed with the LRP algorithm [28], the deconvolution approach [19], and the sensitivity-based method [26] (Section IV-B) to a random order baseline. Exemplary heatmaps produced with these algorithms are displayed and discussed in Section IV-C. At the end of this section we briefly investigate the correlation between heatmap quality and network performance.

A. Setup

We demonstrate the results on a classifier for the MIT Places data set [31] provided by the authors of this data set and the Caffe reference model [40] for ImageNet. We kept the classifiers unchanged. They are both near the state-of-the-art convolutional neural networks and consist of layers of convolution, ReLU, and max-pooling neurons. Both the

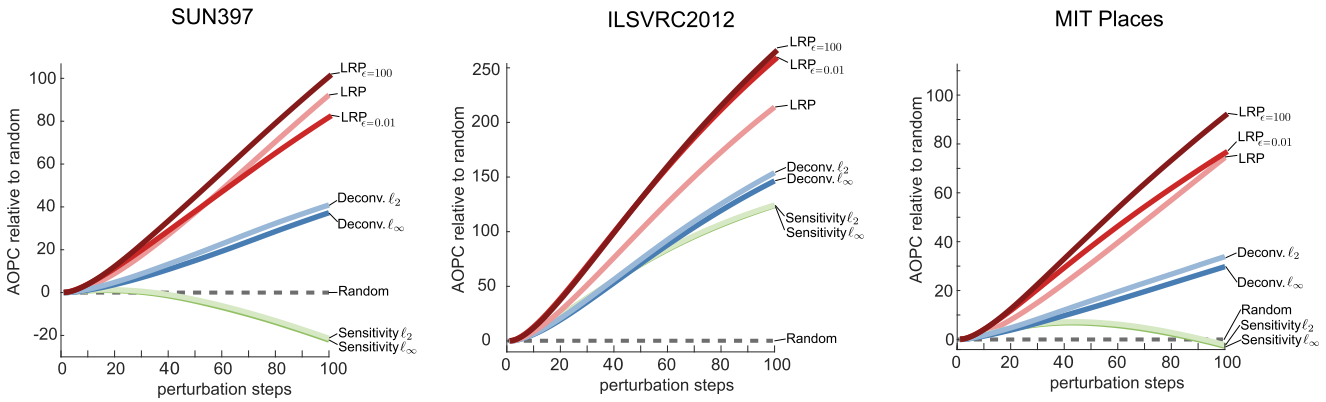


Fig. 4. Comparison of the three heatmapping methods relative to the random baseline. The LRP algorithms have the largest AOPC values, i.e., best explain the classifier's decision, for all the three data sets.

classifiers share the same architecture proposed in [1], namely

$$\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Local Norm} \rightarrow \texttt{Max-Pool}$$
$$\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Local Norm} \rightarrow \texttt{Max-Pool}$$
$$\texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{Conv} \rightarrow \texttt{ReLU}$$
$$\texttt{Max-Pool} \rightarrow \texttt{FC} \rightarrow \texttt{ReLU} \rightarrow \texttt{FC} \rightarrow \texttt{ReLU} \rightarrow \texttt{FC}.$$

The Places classifier was trained on 2 448 873 randomly selected images from 205 categories and the Caffe reference model was trained on 1.2 million images of ImageNet. The MIT Places classifier is used for two testing data sets. First, we compute the AOPC values over 5040 images from the MIT Places testing set. Second, we use AOPC averages over 5040 images from the SUN397 data set [29] as it was done in [41]. We ensured that the category labels of the images used were included in the MIT Places label set. Furthermore, for the ImageNet classifier we report the results on the first 5040 images of the ILSVRC2012 data set. The heatmaps are computed for all the methods for the predicted label, so that our perturbation analysis is a fully unsupervised method during the test stage. Perturbation is applied to 9×9 nonoverlapping regions each covering 0.157% of the image. This region size was selected, because of the following.

1) It allows to perturb a significant part of the image (15.7%) in 100 perturbation steps (for computational reasons we restricted the number of perturbation steps to 100).
2) It approximately matches the size of the convolution filters used by the trained neural network models.

For completeness, the Appendix contains the results for the additional region sizes. We replace all the pixels in a region by randomly sampled (from uniform distribution) values. The choice of a uniform distribution as region perturbation follows one assumption: we consider a region highly relevant if replacing the information in this region in *arbitrary* ways reduces the prediction score of the classifier; we do not want to restrict the analysis to highly specialized information removal schemes. In order to reduce the effect of randomness, we repeat the process ten times. For each ordering we perturb the first 100 regions, resulting for the 9×9 neighborhood in 15.7% of the image being exchanged. Running the experiments for two configurations of perturbations, each with 5040 images, takes roughly 36 h on a workstation with 20 (10×2) Xeon HT-Cores. Given the above running time and the large number of configurations reported here, we considered the choice of 5040 images as sample size a good compromise between the representativeness of our result and the computing time.

B. Quantitative Comparison of Heatmapping Methods

We quantitatively compare the quality of heatmaps generated by the three algorithms described in Section II. As a baseline we also compute the AOPC curves for random heatmaps (i.e., random ordering $\mathcal{O}$). Fig. 4 displays the AOPC values as function of the perturbation steps (i.e., L) relative to the random baseline.

From Fig. 4, one can see that heatmaps computed by LRP have the largest AOPC values, i.e., they better identify the relevant (with respect to the classification tasks) pixels in the image than heatmaps produced with sensitivity analysis or the deconvolution approach. This holds for all the three data sets. The ϵ-LRP formula [see (7)] performs slightly better than α, β-LRP [see (8)], however, we expect both the LRP variants to have a similar performance when optimizing for the parameters (here we use the same settings as in [28]). The deconvolution method performs as the closest competitor and significantly outperforms the random baseline. Since LRP distinguishes between the positive and negative evidence and normalizes the scores properly, it provides less noisy heatmaps than the deconvolution approach (see Section IV-C), which results in a better quantitative performance. As stated above sensitivity analysis targets a slightly different problem and thus provides quantitatively and qualitatively suboptimal explanations of the classifier's decision. Sensitivity provides local explanations, but may fail to capture the global features of a particular class. In this context see also the works of Szegedy *et al.* [21], Goodfellow *et al.* [22], and Nguyen *et al.* [42] in which changing an image as a whole by a minor perturbation leads to a flip in the class labels, and in which rainbow-colored noise images are constructed with high classification accuracy.

The heatmaps computed on the ILSVRC2012 data set are qualitatively better (according to our AOPC measure) than the heatmaps computed on the other two data sets. One reason for this is that the ILSVRC2012 images contain more objects and

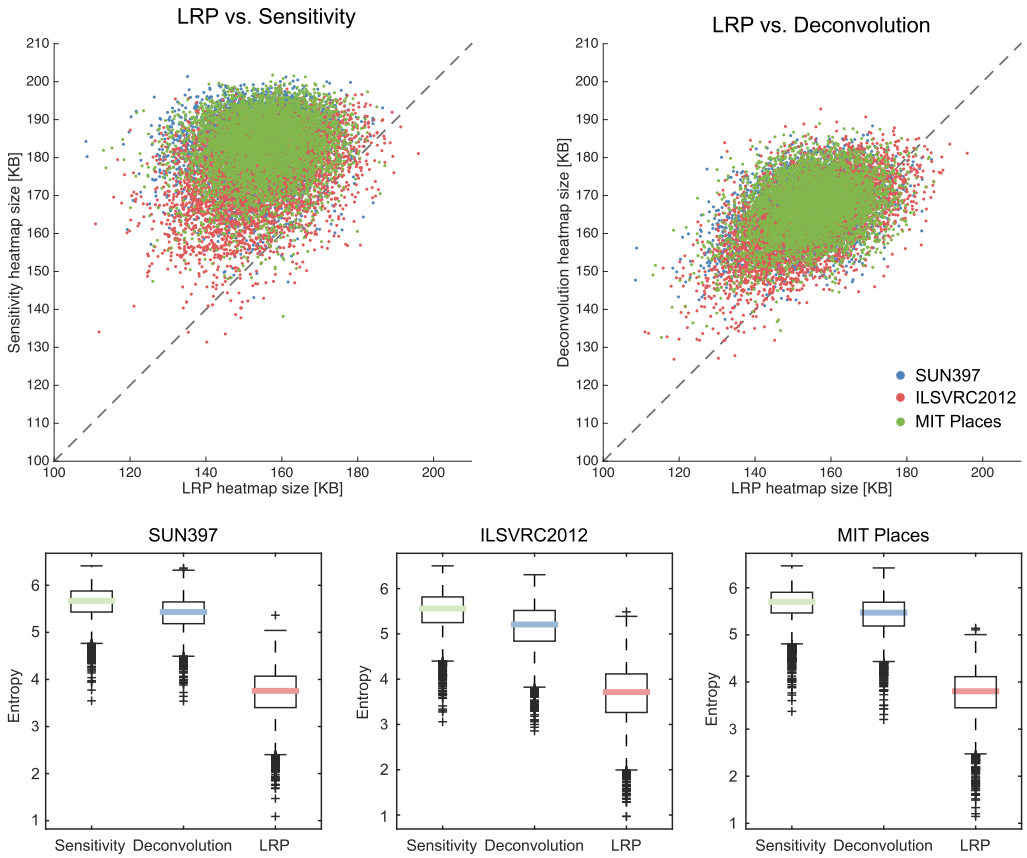


Fig. 5. Comparison of heatmap complexity, measured in terms of file size (top) and image entropy (bottom).

less cluttered scenes than images from the SUN397 and MIT Places data sets, i.e., it is easier (also for humans) to capture the relevant parts of the image. Also, the AOPC difference between the random baseline and the other heatmapping methods is much smaller for the latter two data sets than for ILSVRC2012, because cluttered scenes contain evidence almost everywhere in the image, whereas the background is less important for object categories.

An interesting phenomenon is the performance difference of sensitivity heatmaps computed on SUN397 and MIT Places data sets; in the former case the AOPC curve of sensitivity heatmaps is even below the curve computed with random ranking of regions, whereas for the latter data set the sensitivity heatmaps are (at least initially) clearly better. Note that in both the cases the same classifier [31], trained on the MIT Places data, was used. The difference between these data sets is that SUN397 images lie outside the data manifold (i.e., images of MIT Places used to train the classifier), so that partial derivatives need to explain the local variations of the classification function $f(x)$ in an area in the image space where f has not been trained properly. This effect is not so strong for the MIT Places test data, as they are closer to the images used to train the classifier. Since both LRP and deconvolution provide global explanations, they are less affected by this off-manifold testing.

We performed the above evaluation also for both Caffe networks in the training phase, in which the dropout layers were active. The results are qualitatively the same as the ones shown above. The LRP algorithm, which was explicitly designed to explain the classifier's decision, performs significantly better than the other heatmapping approaches. We would like to stress that LRP does not artificially benefit from the way we evaluate heatmaps as region perturbation is based on an assumption (good heatmaps should rank pixels according to relevance with respect to classification) that is independent of the relevance conservation principle that is used in LRP. Note that LRP was originally designed for binary classifiers in which $f(x) = 0$ denotes the maximal uncertainty about prediction. The classifiers used here were trained with a different multiclass objective, namely, that it suffices for the correct class to have the highest score. One can expect that in such a setup, the state of maximal uncertainty is given by a positive value rather than $f(x) = 0$. In that sense the setup here slightly disfavors LRP. However, we refrained from retraining because it was important for us, first, to use classifiers provided by other researchers in an unmodified manner, and, second, to evaluate the robustness of LRP when applied in the popular multiclass setup.

In addition to perturbation experiments, heatmaps can also be evaluated with respect to their complexity (i.e., sparsity and randomness of the explanations). Good heatmaps highlight the relevant regions and not more, whereas suboptimal heatmaps may contain a number of irrelevant information and noise. In this sense good heatmaps should be better compressible

Fig. 6. Qualitative comparison of the three heatmapping methods for the first eight images of the SUN397, ILSVRC2012, and MIT Places data sets. Red region indicates large scores [see (3) for Sensitivity, (5) for Deconv. and (8) for LRP] while blue region indicates negative scores (only for LRP). The heatmaps computed with the LRP algorithm focus on the relevant features of the object class (e.g., face of the dog or volcano shape), whereas the sensitivity and deconvolution heatmaps are nosier and less focused. These qualitative observations are in line with the quantitative results in Figs. 4 and 5.

than noisy ones. The scatterplots in Fig. 5 show that for almost all the images of the three data sets the LRP heatmap png files are smaller (i.e., less complex) than the corresponding deconvolution and sensitivity files (same holds for jpeg files). Additionally, we report the results obtained using another complexity measure, namely, MATLAB's `entropy` function.

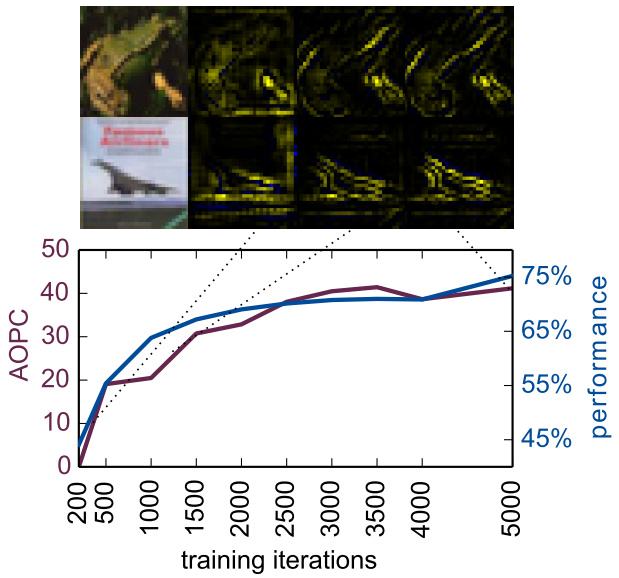


Fig. 7. Evaluation of network performance using AOPC on CIFAR-10.

Also, according to this measure LRP heatmaps are less complex (see boxplots in Fig. 5) than heatmaps computed with the sensitivity and deconvolution methods.

C. Qualitative Comparison of Heatmapping Methods

In Fig. 6, the heatmaps of the first eight images of each data set are visualized. The red region indicates large scores, while the blue region indicates negative scores (only for LRP). The quantitative results presented above are in line with the subjective impressions. The sensitivity and deconvolution heatmaps are noisier and less sparse than the heatmaps computed with the LRP algorithm, reflecting the results obtained in Section IV-B. For SUN397 and MIT Places the sensitivity heatmaps are close to random, whereas both LRP and deconvolution highlight some structural elements in the scene (e.g., mountain shape for the class "volcano" or the arch shape for the class "abbey"). We remark that this bad performance of sensitivity heatmaps does not contradict results like [21] and [22]. In the former works, an image gets modified as a whole, while in this paper, we are considering the quality of selecting local regions and ordering them. Furthermore, gradients are required to move in a very particular direction for reducing the prediction while we are looking for the most relevant regions in the sense that changing them in any way (i.e., randomly) will likely destroy the prediction. The deconvolution and LRP algorithms capture more the global (and more relevant) features than the sensitivity approach.

D. Heatmap Quality and Neural Network Performance

In the last experiment, we briefly show that the quality of a heatmap, as measured by AOPC, provides information about the overall DNN performance. The intuitive explanation for this is that well-trained DNNs capture much better the relevant structures in an image, thus producing more meaningful heatmaps than poorly trained networks that rather rely on the global image statistics. Thus, by evaluating the quality of a heatmap using the proposed procedure, we can potentially assess the network performance, at least for classifiers that were based on the same network topology. Note that this procedure is based on the perturbation of the input of the classifier with the highest predicted score. Thus, this evaluation method is purely unsupervised and does not require labels for the testing images. Fig. 7 depicts the AOPC values and the performance for different training iterations of a DNN for the CIFAR-10 data set [32]. We did not perform these experiments on a larger data set since the effect can still be observed nicely in this modest data size. The correlation between both the curves indicates that heatmaps contain information that can potentially be used to judge the quality of the network. This paper did not intend to profoundly investigate the relation between network performance and heatmap quality, which is a topic for future research.

V. Conclusion

Research in DNN has been traditionally focusing on improving the quality, algorithmics, or the speed of a neural network model. We have studied an orthogonal research direction in our manuscript, namely, we have contributed to furthering the understanding and transparency of the decision making implemented by a trained DNN: for this we have focused on the heatmap concept that, e.g., in a computer vision application, is able to attribute the contribution of individual pixels to the DNN inference result for a novel data sample. While heatmaps allow better intuition about what has been learned by the network, we tackled the so far open problem of quantifying the quality of a heatmap. In this manner, different heatmap algorithms can be compared quantitatively and their properties and limits can be related. We proposed a region perturbation strategy that is based on the idea that flipping the most salient pixels first should lead to high performance decay. A large AOPC value as a function of the perturbation steps was shown to provide a good measure for a very informative heatmap. We also showed quantitatively and qualitatively that sensitivity maps and heatmaps computed with the deconvolution algorithm are much noisier than heatmaps computed with the LRP method, and thus are less suitable for identifying the most important regions with respect to the classification task. Above all, we provided first evidence that heatmaps may be useful for assessment of the neural network performance. Bringing this idea into practical application will be a topic of the future research. Concluding, we have provided the basis for an accurate quantification of heatmap quality.

Note that a good heatmap can not only be used for better understanding of DNNs but also for a prioritization of the image regions. Thus, the regions of an individual image with high heatmap values could be subjected to a more detailed analysis. This could in the future allow highly time efficient processing of the data only *where it matters*.

Appendix

We performed the perturbation experiments from Section IV-B also with region sizes smaller and larger

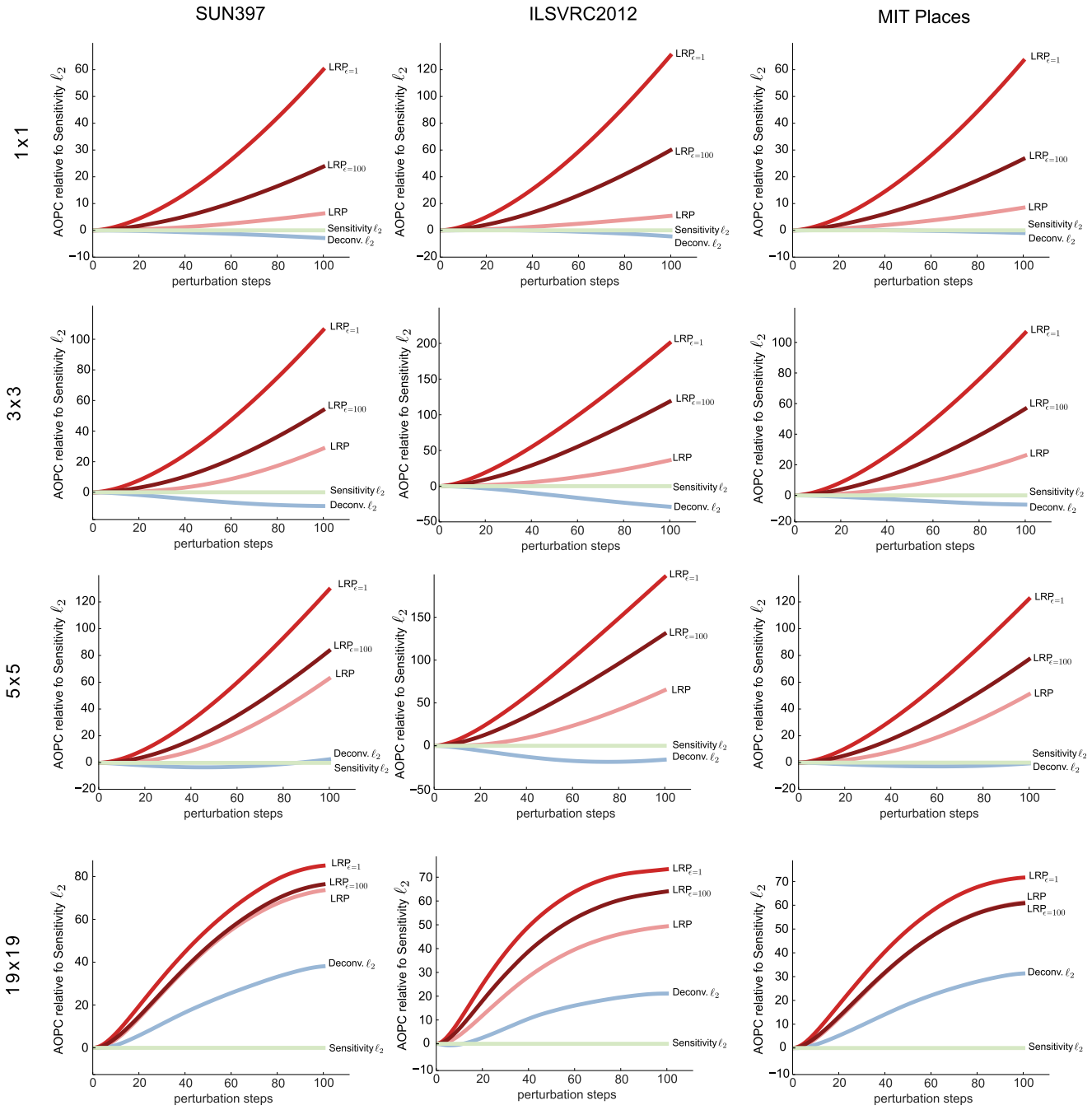


Fig. 8. Comparison of the three heatmapping methods relative to the sensitivity ℓ_2 baseline for different region sizes.

than 9×9, namely, 1×1, 3×3, 5×5 and 19×19. Fig. 8 shows the results relative to the sensitivity-ℓ_2 baseline. Due to the high computational load, we did not compute the random baseline and also left out the ℓ_∞ norm as it performed similarly to ℓ_2. Since LRP$_{\epsilon=100}$ performed much better than LRP$_{\epsilon=0.01}$ in Section IV-B, we decided to leave out the latter and to include an LRP variant with an ϵ value between 0.01 and 100. We decided to take $\epsilon = 1$.

Qualitatively, the results are very similar to that reported in Fig. 4 for the 9×9 region size. In all the cases, the ϵ-stabilized LRP algorithm provides the best explanations. Interestingly, $\epsilon = 1$ outperforms the results obtained with significantly larger or smaller stabilizing factors. Thus although stabilization is important (as discussed in Section IV-B), too large ϵ values may decrease the efficiency of LRP as it suppresses contributions from small relevances. Another observation that can be made from Fig. 8 is that the deconvolution method outperforms sensitivity analysis in terms of AOPC only for large region sizes (9×9 in Fig. 4 and 19×19 in Fig. 8). Note that the explanations computed by deconvolution show a large number of responses that look like filter visualizations (see Fig. 6). The kernel size for the used networks is 11. We argue that with a region size close to or larger than the kernel size, we are obtaining a score of the region that is close to the score of the filter as a whole. Destroying such a region has a large impact on the filter response and thus

leads to a decrease in the classification score. Destroying individual pixels (or small regions) on the other hand only slightly affects the filter response and results in small AOPC values. LRP performs well for all the region sizes, because it does not focus on filter responses but identifies the relevant pixels in the image. Note that these pixels often represent the shape of the object (see Fig. 6), so that destroying the small regions around these pixels largely destroys the object shape that leads to a fast decrease in the classification score.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. 2012, pp. 1097–1105.

[2] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Adv. NIPS*, 2012, pp. 2852–2860.

[3] C. Szegedy *et al.* (2014). "Going deeper with convolutions." [Online]. Available: http://arxiv.org/abs/1409.4842

[4] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE CVPR*, Jun. 2012, pp. 3642–3649.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011.

[6] R. Socher *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, 2013, pp. 1631–1642.

[7] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," in *Proc. ICML*, 2010, pp. 495–502.

[8] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. CVPR*, 2011, pp. 3361–3368.

[9] G. Montavon *et al.*, "Machine learning of molecular electronic properties in chemical compound space," New J. Phys., vol. 15, no. 9, p. 095003, 2013.

[10] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade* (Lecture Notes in Computer Science), vol. 7700, 2nd ed. Berlin, Germany: Springer, 2012.

[11] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–50.

[12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[13] M. L. Braun, J. M. Buhmann, and K.-R. Müller, "On relevant dimensions in kernel feature spaces," *J. Mach. Learn. Res.*, vol. 9, pp. 1875–1908, Aug. 2008.

[14] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, pp. 1803–1831, Mar. 2010.

[15] K. Hansen, D. Baehrens, T. Schroeter, M. Rupp, and K.-R. Müller, "Visual interpretation of kernel-based prediction models," *Molecular Inform.*, vol. 30, no. 9, pp. 817–826, 2011.

[16] G. Montavon, M. L. Braun, T. Krueger, and K.-R. Müller, "Analyzing local structure in kernel-based learning: Explanation, complexity, and reliability assessment," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 62–74, Jul. 2013.

[17] D. Erhan, A. Courville, and Y. Bengio, "Understanding representations learned in deep architectures," Dépt. d'Informatique Recherche Opérationnelle, Univ. Montréal, Montreal, QC, Canada, Tech. Rep. 1355, 2010.

[18] G. Montavon, M. L. Braun, and K.-R. Müller, "Kernel analysis of deep networks," *J. Mach. Learn. Res.*, vol. 12, pp. 2563–2581, Sep. 2011.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[20] A. Mahendran and A. Vedaldi. (2014). "Understanding deep image representations by inverting them." [Online]. Available: http://arxiv.org/abs/1412.0035

[21] C. Szegedy *et al.* (2013). "Intriguing properties of neural networks." [Online]. Available: http://arxiv.org/abs/1312.6199

[22] I. J. Goodfellow, J. Shlens, and C. Szegedy. (2014). "Explaining and harnessing adversarial examples." [Online]. Available: http://arxiv.org/abs/1412.6572

[23] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. (2015). "Understanding neural networks through deep visualization." [Online]. Available: http://arxiv.org/abs/1506.06579

[24] A. Dosovitskiy and T. Brox. (2016). "Inverting visual representations with convolutional networks." [Online]. Available: http://arxiv.org/abs/1506.02753

[25] P. M. Rasmussen *et al.*, "Visualization of nonlinear classification models in neuroimaging—Signed sensitivity maps," in *Proc. BIOSIGNALS*, 2012, pp. 254–263.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. ICLR Workshop*, 2014, pp. 1–8.

[27] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. IEEE CVPR*, 2016, pp. 2912–2920.

[28] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLOS ONE*, vol. 10, no. 7, p. e0130140, 2015.

[29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. CVPR*, 2010, pp. 3485–3492.

[30] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Adv. NIPS*, 2014, pp. 487–495.

[32] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Univ. Toronto, Toronto, ON, Canada, 2009.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[34] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Müller, and W. Samek, "The LRP toolbox for artificial neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 114, pp. 1–5, 2016.

[35] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller. (2015). "Explaining nonlinear classification decisions with deep Taylor decomposition." [Online]. Available: http://arxiv.org/abs/1512.02479

[36] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vis. Res.*, vol. 40, nos. 10–12, pp. 1489–1506, Jun. 2000.

[37] D. J. Heeger, E. P. Simoncelli, and J. A. Movshon, "Computational models of cortical visual processing," *Proc. Nat. Acad. Sci. USA*, vol. 93, no. 2, pp. 623–627, 1996.

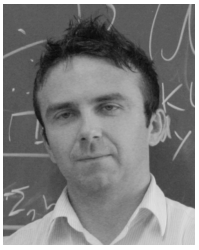
[38] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annu. Rev. Neurosci.*, vol. 24, no. 1, pp. 1193–1216, 2001.

[39] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, 2001.

[40] Y. Jia. (2013). *Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding*. [Online]. Available: http://caffe.berkeleyvision.org/

[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. (2014). "Object detectors emerge in deep scene CNNs." [Online]. Available: http://arxiv.org/abs/1412.6856

[42] A. Nguyen, J. Yosinski, and J. Clune. (2014). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." [Online]. Available: http://arxiv.org/abs/1412.1897



Wojciech Samek (M'13) received the Diploma degree in computer science from the Humboldt University of Berlin, Berlin, Germany, in 2010, and the Ph.D. degree in machine learning from the Technische Universität Berlin, Berlin, in 2014.

He visited Heriot–Watt University, Edinburgh, U.K., and the University of Edinburgh, Edinburgh, from 2007 to 2008. In 2009, he was with the Intelligent Robotics Group, NASA Ames Research Center, Mountain View, CA, USA. In 2014, he founded the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, Berlin, where he is currently the Director. He was a Scholar of the German National Academic Foundation, and a Ph.D. Fellow at the Bernstein Center for Computational Neuroscience Berlin, Berlin. He is associated with the Berlin Big Data Center. His current research interests include interpretable machine learning, neural networks, robust signal processing, and computer vision.

Alexander Binder (M'15) received the Diploma degree in mathematics from the Humboldt University of Berlin, Berlin, Germany, and the Ph.D. degree from the Department of Computer Science, Technische Universität Berlin, Berlin, in 2013.

He has been involved in the THESEUS project on semantic image retrieval at Fraunhofer FIRST, Berlin, since 2007, where he was the Principal Contributor to top five ranked submissions at ImageCLEF and Pascal VOC challenges. He has been an Assistant Professor with the Singapore University of Technology and Design, Singapore, since 2015. His current research interests include computer vision, explanations of deep neural nets, medical applications, machine learning, and efficient heuristics.

Grégoire Montavon received the master's degree in communication systems from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2009, and the Ph.D. degree in machine learning from the Technische Universität Berlin (TU Berlin), Berlin, in 2013.

He is currently a Research Associate with the Machine Learning Group, TU Berlin. His current research interests include neural networks, machine learning, and data analysis.

Sebastian Lapuschkin received the master's degree in computer science from Technische Universität Berlin, Berlin, Germany, in 2013, where he is currently pursuing the Ph.D. degree.

He is currently a Research Associate with the Machine Learning Group, Fraunhofer Heinrich Hertz Institute, Berlin. His current research interests include computer vision, machine learning, and data analysis.

Klaus-Robert Müller (M'12) received the Degree in physics and the Ph.D. degree in computer science from the Technische Universität Karlsruhe, Karlsruhe, Germany, in 1989 and 1992, respectively.

He has been a Professor of Computer Science at Technische Universität Berlin, Berlin, Germany, since 2006, as well as the Co-Director of the Berlin Big Data Center. He held a post-doctoral position at Fraunhofer FIRST, Berlin. He was a Research Fellow with the University of Tokyo, Tokyo, Japan, from 1994 to 1995. In 1995, he founded the Intelligent Data Analysis Group at Fraunhofer FIRST and was the Director until 2008. From 1999 to 2006, he was a Professor at the University of Potsdam, Potsdam, Germany. His current research interests include intelligent data analysis, machine learning, signal processing, and brain-computer interfaces.

Prof. Müller was a recipient of the 1999 Olympus Prize by the German Pattern Recognition Society, DAGM, in 2006, he received the SEL Alcatel Communication Award, and in 2014, he received the Science Prize of Berlin awarded by the Governing Mayor of Berlin. In 2012, he was elected as a member of the German National Academy of Sciences–Leopoldina.