



Interpretability of Machine Learning Methods Applied to Neuroimaging

Elina Thibeau-Sutre, Sasha Collin, Ninon Burgos, and Olivier Colliot

Abstract

Deep learning methods have become very popular for the processing of natural images and were then successfully adapted to the neuroimaging field. As these methods are non-transparent, interpretability methods are needed to validate them and ensure their reliability. Indeed, it has been shown that deep learning models may obtain high performance even when using irrelevant features, by exploiting biases in the training set. Such undesirable situations can potentially be detected by using interpretability methods. Recently, many methods have been proposed to interpret neural networks. However, this domain is not mature yet. Machine learning users face two major issues when aiming to interpret their models: which method to choose and how to assess its reliability. Here, we aim at providing answers to these questions by presenting the most common interpretability methods and metrics developed to assess their reliability, as well as their applications and benchmarks in the neuroimaging context. Note that this is not an exhaustive survey: we aimed to focus on the studies which we found to be the most representative and relevant.

Key words Interpretability, Saliency, Machine learning, Deep learning, Neuroimaging, Brain disorders

1 Introduction

1.1 *Need for Interpretability*

Many metrics have been developed to evaluate the performance of machine learning (ML) systems. In the case of supervised systems, these metrics compare the output of the algorithm to a ground truth, in order to evaluate its ability to reproduce a label given by a physician. However, the users (patients and clinicians) may want more information before relying on such systems. On which features is the model relying to compute the results? Are these features close to the way a clinician thinks? If not, why? This questioning coming from the actors of the medical field is justified, as errors in real life may lead to dramatic consequences. Trust into ML systems cannot be built only based on a set of metrics evaluating the performance of the system. Indeed, various examples of machine learning systems taking correct decisions for the wrong reasons

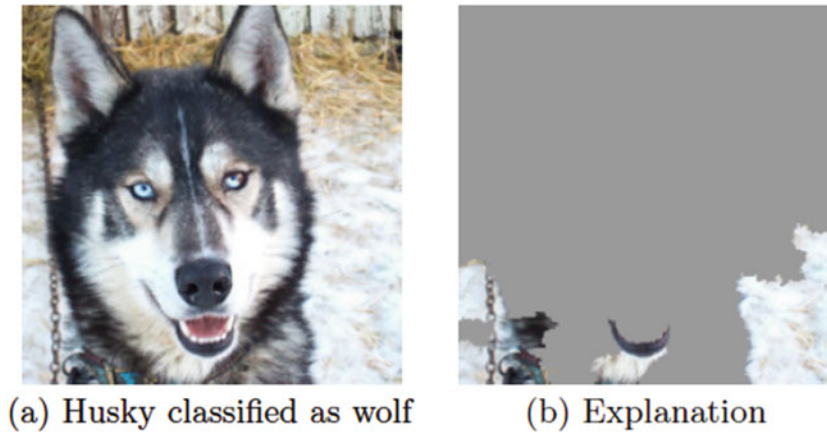


Fig. 1 Example of an interpretability method highlighting why a network took the wrong decision. The explained classifier was trained on the binary task “Husky” vs “Wolf.” The pixels used by the model are actually in the background and highlight the snow. (Adapted from [1]. Permission to reuse was kindly granted by the authors)

exist, e.g., [1–3]. Thus, even though their performance is high, they may be unreliable and, for instance, not generalize well to slightly different data sets. One can try to prevent this issue by interpreting the model with an appropriate method whose output will highlight the reasons why a model took its decision.

In [1], the authors show a now classical case of a system that correctly classifies images for wrong reasons. They purposely designed a biased data set in which wolves always are in a snowy environment whereas huskies are not. Then, they trained a classifier to differentiate wolves from huskies: this classifier had good accuracy but classified wolves as huskies when the background was snowy and huskies as wolves when there was no snow. Using an interpretability method, they further highlighted that the classifier was looking at the background and not at the animal (*see* Fig. 1).

Another study [2] detected a bias in ImageNet (a widely used data set of natural images) as the interpretation of images with the label “chocolate sauce” highlighted the importance of the spoon. Indeed, ImageNet “chocolate sauce” images often contained spoons, leading to a spurious correlation. There are also examples of similar problems in medical applications. For instance, a recent paper [3] showed with interpretability methods that some deep learning systems detecting COVID-19 from chest radiographs actually relied on confounding factors rather than on the actual pathological features. Indeed, their model focused on other regions than the lungs to evaluate the COVID-19 status (edges, diaphragm, and cardiac silhouette). Of note, their model was trained on public data sets which were used by many studies.

1.2 How to Interpret Models

According to [4], model interpretability can be broken down into two categories: transparency and post hoc explanations.

A model can be considered as transparent when it (or all parts of it) can be fully understood as such, or when the learning process is understandable. A natural and common candidate that fits, at first sight, these criteria is the linear regression algorithm, where coefficients are usually seen as the individual contributions of the input features. Another candidate is the decision tree approach where model predictions can be broken down into a series of understandable operations. One can reasonably consider these models as transparent: one can easily identify the features that were used to take the decision. However, one may need to be cautious not to push too far the medical interpretation. Indeed, the fact that a feature has not been used by the model does not mean that it is not associated with the target. It just means that the model did not need it to increase its performance. For instance, a classifier aiming at diagnosing Alzheimer's disease may need only a set of regions (for instance, from the medial temporal lobe of the brain) to achieve an optimal performance. This does not mean that other brain regions are not affected by the disease, just that they were not used by the model to take its decision. This is the case, for example, for sparse models like LASSO, but also standard multiple linear regressions. Moreover, features given as input to transparent models are often highly engineered, and choices made before the training step (preprocessing, feature selection) may also hurt the transparency of the whole framework. Nevertheless, in spite of these caveats, such models can reasonably be considered transparent, in particular when compared to deep neural networks which are intrinsically black boxes.

The second category of interpretability methods, post hoc interpretations, allows dealing with non-transparent models. Xie et al. [5] proposed a taxonomy in three categories: *visualization* methods consist in extracting an attribution map of the same size as the input whose intensities allow knowing where the algorithm focused its attention, *distillation* approaches consist in reproducing the behavior of a black box model with a transparent one, and *intrinsic* strategies include interpretability components within the framework, which are trained along with the main task (e.g., a classification). In the present work, we focus on this second category of methods (post hoc) and proposed a new taxonomy including other methods of interpretation (see Fig. 2). Post hoc interpretability is the most used category nowadays, as it allows interpreting deep learning methods that became the state of the art for many tasks in neuroimaging, as in other application fields.

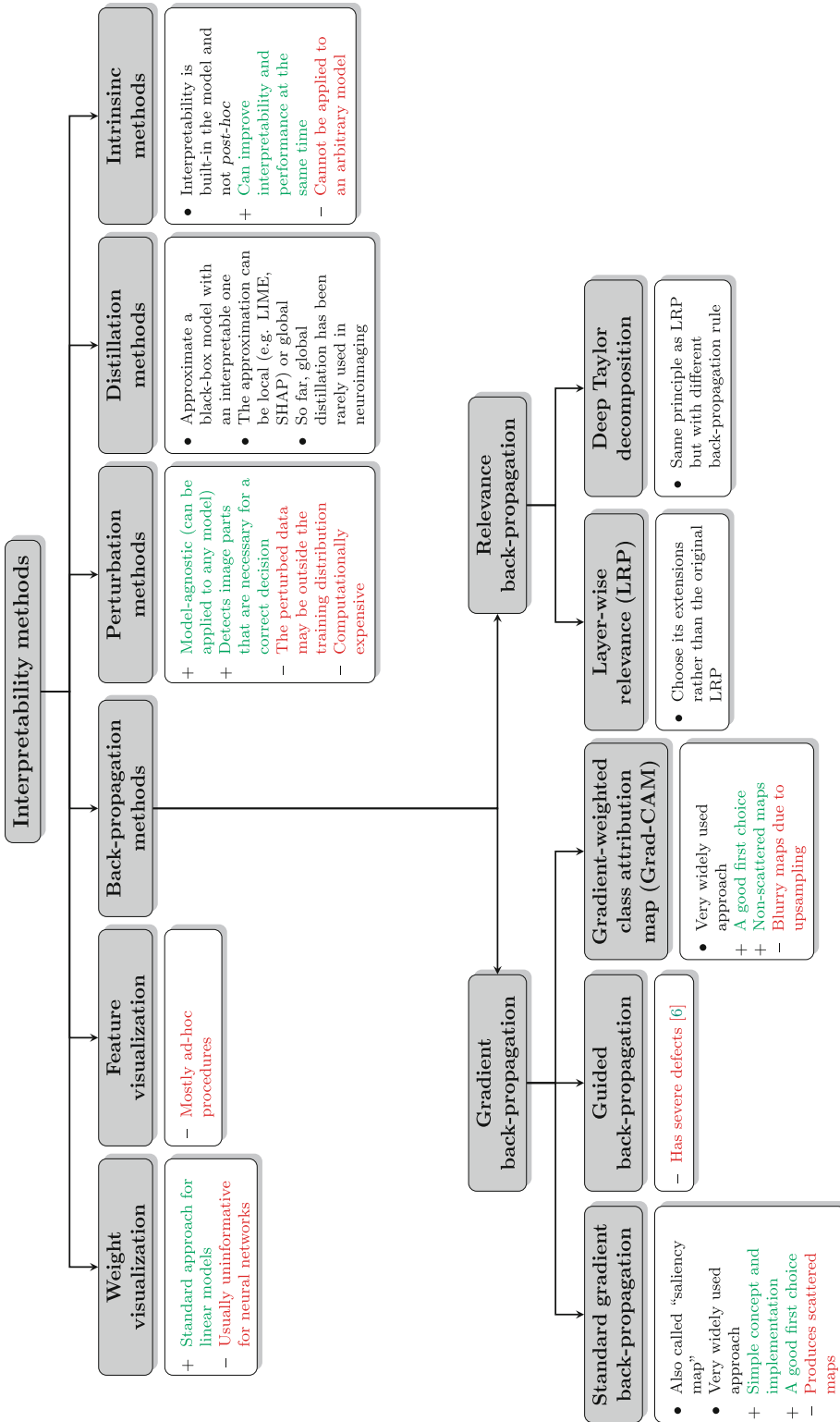


Fig. 2 Taxonomy of the main interpretability methods

1.3 Chapter Content and Outline

This chapter focuses on methods developed to interpret non-transparent machine learning systems, mainly deep learning systems, computing classification, or regression tasks from high-dimensional inputs. The interpretability of other frameworks (in particular generative models such as variational autoencoders or generative adversarial networks) is not covered as there are not enough studies addressing them. It may be because high-dimensional outputs (such as images) are easier to interpret “as such,” whereas small dimensional outputs (such as scalars) are less transparent.

Most interpretability methods presented in this chapter produce an attribution map: an array with the same dimensions as that of the input (up to a resizing) that can be overlaid on top of the input in order to exhibit an explanation of the model prediction. In the literature, many different terms may coexist to name this output such as saliency map, interpretation map, or heatmap. To avoid misunderstandings, in the following, we will only use the term “attribution map.”

The chapter is organized as follows. Subheading 2 presents the most commonly used interpretability methods proposed for computer vision, independently of medical applications. It also describes metrics developed to evaluate the reliability of interpretability methods. Then, Subheading 3 details their application to neuroimaging. Finally, Subheading 4 discusses current limitations of interpretability methods, presents benchmarks conducted in the neuroimaging field, and gives some advice to the readers who would like to interpret their own models.

Mathematical notations and abbreviations used during this chapter are summarized in Tables 1 and 2. A short reminder on neural network training procedure and a brief description of the diseases mentioned in the present chapter are provided in Appendices A and B.

2 Interpretability Methods

This section presents the main interpretability methods proposed in the domain of computer vision. We restrict ourselves to the methods that have been applied to the neuroimaging domain (the applications themselves being presented in Subheading 3). The outline of this section is largely inspired from the one proposed by Xie et al. [5]:

1. **Weight visualization** consists in directly visualizing weights learned by the model, which is natural for linear models but quite less informative for deep learning networks.

2. **Feature map visualization** consists in displaying intermediate results produced by a deep learning network to better understand its operation principle.
3. **Back-propagation methods** back-propagate a signal through the machine learning system from the output node of interest o_c to the level of the input to produce an attribution map.

Table 1
Mathematical notations

X_0 is the input tensor given to the network, and X refers to any input, sampled from the set \mathcal{X} .
y is a vector of target classes corresponding to the input.
f is a network of L layers. The first layer is the closest to the input; the last layer is the closest to the output. A layer is a function.
g is a transparent function which aims at reproducing the behavior of f .
w and b are the weights and the bias associated to a linear function (e.g., in a fully connected layer).
u and v are locations (set of coordinates) corresponding to a node in a feature map. They belong respectively to the set \mathcal{U} and \mathcal{V} .
$A_k^{(l)}(u)$ is the value of the feature map computed by layer l , of K channels at channel k , at position u .
$R_k^{(l)}(u)$ is the value of a property back-propagated through the $l+1$, of K channels at channel k , at position u . $R^{(l)}$ and $A^{(l)}$ have the same number of channels.
o_c is the output node of interest (in a classification framework, it corresponds to the node of the class c).
S_c is an attribution map corresponding to the output node o_c .
m is a mask of perturbations. It can be applied to X to compute its perturbed version X^m .
Φ is a function producing a perturbed version of an input X .
Γ_c is the function computing the attribution map S_c from the black-box function f and an input X_0 .

Table 2
Abbreviations

CAM Class activation maps
CNN Convolutional neural network
CT Computed tomography
Grad-CAM Gradient-weighted class activation mapping
LIME Local interpretable model-agnostic explanations
LRP Layer-wise relevance
MRI Magnetic resonance imaging
SHAP SHapley Additive exPlanations
T1w T1-weighted [Magnetic Resonance Imaging]

4. **Perturbation methods** locally perturb the input and evaluate the difference in performance between using the original input and the perturbed version to infer which parts of the input are relevant for the machine learning system.
5. **Distillation** approximates the behavior of a black box model with a more transparent one and then draw conclusions from this new model.
6. **Intrinsic** includes the only methods of this chapter that are not post hoc explanations: in this case, interpretability is obtained thanks to components of the framework that are trained at the same time as the model.

Finally, for the methods producing an attribution map, a section is dedicated to the metrics used to evaluate different properties (e.g., reliability or human intelligibility) of the maps.

We caution readers that this taxonomy is not perfect: some methods may belong to several categories (e.g., LIME and SHAP could belong either to perturbation or distillation methods). Moreover, interpretability is still an active research field, and then some categories may (dis)appear or be fused in the future.

The interpretability methods were (most of the time) originally proposed in the context of a classification task. In this case, the network outputs an array of size C , corresponding to the number of different labels existing in the data set, and the goal is to know how the output node corresponding to a particular class c interacts with the input or with other parts of the network. However, these techniques can be extended to other tasks: for example, for a regression task, we will just have to consider the output node containing the continuous variable learned by the network. Moreover, some methods do not depend on the nature of the algorithm (e.g., standard perturbation or LIME) and can be applied to any machine learning algorithm.

2.1 *Weight Visualization*

At first sight, one of can be tempted to directly visualize the weights learned by the algorithm. This method is really simple, as it does not require further processing. However, even though it can make sense for linear models, it is not very informative for most networks unless they are specially designed for this interpretation.

This is the case for AlexNet [7], a convolutional neural network (CNN) trained on natural images (ImageNet). In this network the size of the kernels in the first layer is large enough (11×11) to distinguish patterns of interest. Moreover, as the three channels in the first layer correspond to the three color channels of the images (red, green, and blue), the values of the kernels can also be represented in terms of colors (this is not the case for hidden layers, in which the meaning of the channels is lost). The 96 kernels of the first layer were illustrated in the original article as in Fig. 3.

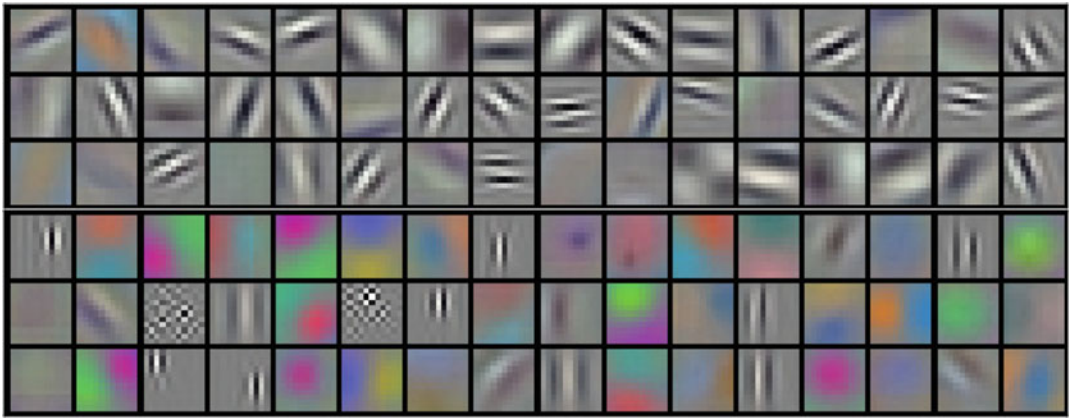


Fig. 3 96 convolutional kernels of size $3@11 \times 11$ learned by the first convolutional layer on the $3@224 \times 224$ input images by AlexNet. (Adapted from [7]. Permission to reuse was kindly granted by the authors)

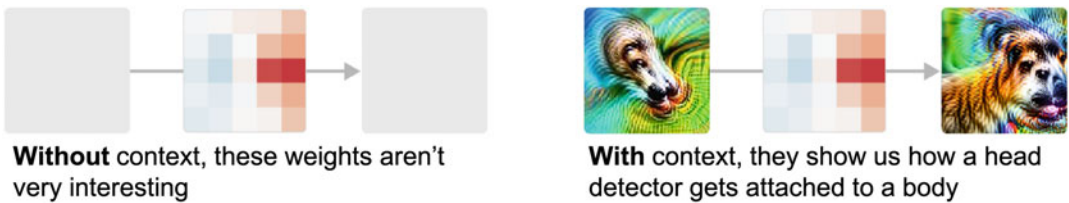


Fig. 4 The weights of small kernels in hidden layers (here 5×5) can be really difficult to interpret alone. Here some context allows better understanding how it modulates the interaction between concepts conveyed by the input and the output. (Adapted from [8] (CC BY 4.0))

However, for hidden layers, this kind of interpretation may be misleading as nonlinearity activation layers are added between the convolutions and fully connected layers; this is why they only visualized the weights of the first layer.

To understand the weight visualization in hidden layers of a network, Voss et al. [8] proposed to add some context to the input and the output channels. This way they enriched the weight visualization with feature visualization methods able to generate an image corresponding to the input node and the output node (see Fig. 4). However, the feature visualization methods used to bring some context can also be difficult to interpret themselves, and then it only moves the interpretability problem from weights to features.

2.2 Feature Map Visualization

Feature maps are the results of intermediate computations done from the input and resulting in the output value. Then, it seems natural to visualize them or link them to concepts to understand how the input is successively transformed into the output.

Methods described in this section aim at highlighting which concepts a feature map (or part of it) A conveys.

2.2.1 Direct Interpretation

The output of a convolution has the same shape as its input: a 2D image processed by a convolution will become another 2D image (the size may vary). Then, it is possible to directly visualize these feature maps and compare them to the input to understand the operations performed by the network. However, the number of filters of convolutional layers (often a hundred) makes the interpretation difficult as a high number of images must be interpreted for a single input.

Instead of directly visualizing the feature map A , it is possible to study the latent space including all the values of the samples of a data set at the level of the feature map A . Then, it is possible to study the deformations of the input by drawing trajectories between samples in this latent space, or more simply to look at the distribution of some label in a manifold learned from the latent space. In such a way, it is possible to better understand which patterns were detected, or at which layer in the network classes begin to be separated (in the classification case). There is often no theoretical framework to illustrate these techniques, and then we referred to studies in the context of the medical application (*see* Subheading 3.2 for references).

2.2.2 Input Optimization

Olah et al. [9] proposed to compute an input that maximizes the value of a feature map A (*see* Fig. 5). However, this technique leads to unrealistic images that may be themselves difficult to interpret, particularly for neuroimaging data. To have a better insight of the behavior of layers or filters, another simple technique illustrated by the same authors consists in isolating the inputs that led to the highest activation of A . The combination of both methods, displayed in Fig. 6, allows a better understanding of the concepts conveyed by A of a GoogLeNet trained on natural images.

2.3 Back-Propagation Methods

The goal of these interpretability methods is to link the value of an output node of interest o_c to the image X_0 given as input to a network. They do so by back-propagating a signal from o_c to X_0 :

Different **optimization objectives** show what different parts of a network are looking for.

n layer index
x, y spatial position
z channel index
k class index

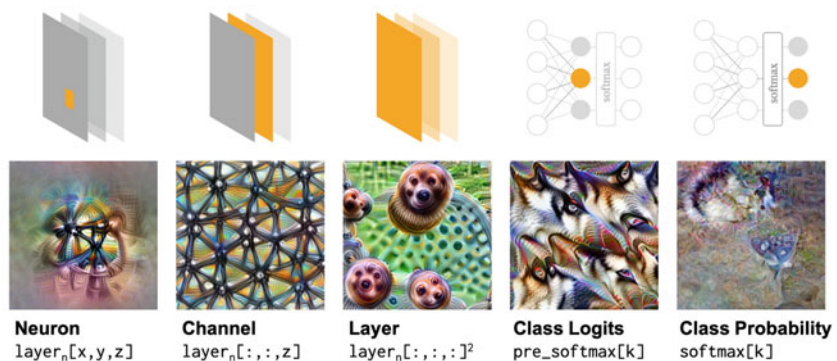


Fig. 5 Optimization of the input for different levels of feature maps. (Adapted from [9] (CC BY 4.0))

Dataset Examples show us what neurons respond to in practice



Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes? *mixed4a, Unit 6* Animal faces—or snouts? *mixed4a, Unit 240* Clouds—or fluffiness? *mixed4a, Unit 453* Buildings—or sky? *mixed4a, Unit 492*

Fig. 6 Interpretation of a neuron of a feature map by optimizing the input associated with a bunch of training examples maximizing this neuron. (Adapted from [9] (CC BY 4.0))

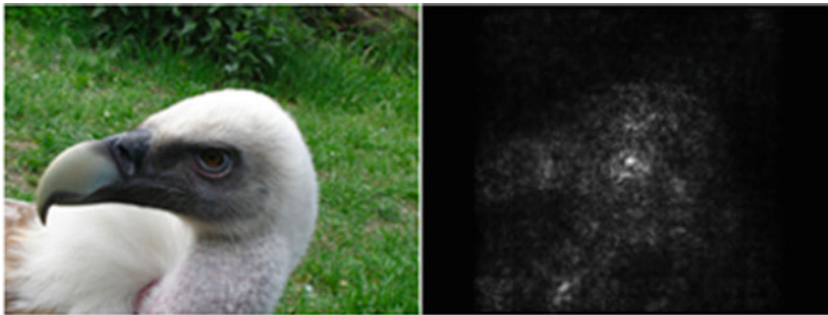


Fig. 7 Attribution map of an image found with gradients back-propagation. (Adapted from [10]. Permission to reuse was kindly granted by the authors)

this process (backward pass) can be seen as the opposite operation than the one done when computing the output value from the input (forward pass).

Any property can be back-propagated as soon as its value at the level of a feature map $l - 1$ can be computed from its value in the feature map l . In this section, the back-propagated properties are gradients or the relevance of a node o_c .

2.3.1 Gradient Back-Propagation

During network training, gradients corresponding to each layer are computed according to the loss to update the weights. Then, we can see these gradients as the difference needed at the layer level to improve the final result: by adding this difference to the weights, the probability of the true class y increases.

In the same way, the gradients can be computed at the image level to find how the input should vary to change the value of o_c (see example on Fig. 7). This gradient computation was proposed by [10], in which the attribution map S_c corresponding to the input

image X_0 and the output node o_c is computed according to the following equation:

$$S_c = \frac{\partial o_c}{\partial X} \Big|_{X=X_0} \quad (1)$$

Due to its simplicity, this method is the most commonly used to interpret deep learning networks. Its attribution map is often called a “saliency map”; however, this term is also used in some articles to talk about any attribution map, and this is why we chose to avoid this term in this chapter.

This method was modified to derive many similar methods based on gradient computation described in the following paragraphs.

Gradient \odot Input This method is the point-wise product of the gradient map described at the beginning of the section and the input. Evaluated in [11], it was presented as an improvement of the gradients method, though the original paper does not give strong arguments on the nature of this improvement.

DeconvNet & Guided Back-Propagation The key difference between this procedure and the standard back-propagation method is the way the gradients are back-propagated through the ReLU layer.

The ReLU layer is a commonly used activation function that sets to 0 the negative input values and does not affect positive input values. The derivative of this function in layer l is the indicator function $\mathbb{1}_{A^{(l)} > 0}$: it outputs 1 (resp. 0) where the feature maps computed during the forward pass were positive (resp. negative).

Springenberg et al. [12] proposed to back-propagate the signal differently. Instead of applying the indicator function of the feature map $A^{(l)}$ computed during the forward pass, they directly applied ReLU to the back-propagated values $R^{(l+1)} = \frac{\partial o_c}{\partial A^{(l+1)}}$, which corresponds to multiplying it by the indicator function $\mathbb{1}_{R^{(l+1)} > 0}$. This “backward deconvnet” method allows back-propagating only the positive gradients, and, according to the authors, it results in a reconstructed image showing the part of the input image that is most strongly activating this neuron.

The guided back-propagation method (Eq. 4) combines the standard back-propagation (Eq. 2) with the backward deconvnet (Eq. 3): when back-propagating gradients through ReLU layers, a value is set to 0 if the corresponding top gradients or bottom data is negative. This adds an additional guidance to the standard back-propagation by preventing backward flow of negative gradients.

$$R^{(l)} = \mathbb{1}_{A^{(l)} > 0} * R^{(l+1)} \tag{2}$$

$$R^{(l)} = \mathbb{1}_{R^{(l+1)} > 0} * R^{(l+1)} \tag{3}$$

$$R^{(l)} = \mathbb{1}_{A^{(l)} > 0} * \mathbb{1}_{R^{(l+1)} > 0} * R^{(l+1)} \tag{4}$$

Any back-propagation procedure can be “guided,” as it only concerns the way ReLU functions are managed during back-propagation (this is the case, e.g., for guided Grad-CAM).

While it was initially adopted by the community, this method showed severe defects as discussed later in Subheading 4.

CAM & Grad-CAM In this setting, attribution maps are computed at the level of a feature map produced by a convolutional layer and then upsampled to be overlaid and compared with the input. The first method, class activation maps (CAM), was proposed by Zhou et al. [13] and can be only applied to CNNs with the following specific architecture:

1. A series of convolutions associated with activation functions and possibly pooling layers. These convolutions output a feature map A with N channels.
2. A global average pooling that extracts the mean value of each channel of the feature map produced by the convolutions.
3. A single fully connected layer

The CAM corresponding to o_c will be the mean of the channels of the feature map produced by the convolutions, weighted by the weights w_{kc} learned in the fully connected layer

$$S_c = \sum_{k=1}^N w_{kc} * A_k. \tag{5}$$

This map has the same size as A_k , which might be smaller than the input if the convolutional part performs downsampling operations (which is very often the case). Then, the map is upsampled to the size of the input to be overlaid on the input.

Selvaraju et al. [14] proposed an extension of CAM that can be applied to any architecture: Grad-CAM (illustrated on Fig. 8). As in CAM, the attribution map is a linear combination of the channels of a feature map computed by a convolutional layer. But, in this case, the weights of each channel are computed using gradient back-propagation

$$\alpha_{kc} = \frac{1}{|U|} \sum_{u \in U} \frac{\partial o_c}{\partial A_k(u)}. \tag{6}$$



Fig. 8 Grad-CAM explanations highlighting two different objects in an image. (a) the original image, (b) the explanation based on the “dog” node, (c) the explanation based on the “cat” node. ©2017 IEEE. (Reprinted, with permission, from [14])

The final map is then the linear combination of the feature maps weighted by the coefficients. A ReLU activation is then applied to the result to only keep the features that have a positive influence on class c

$$S_c = \text{ReLU} \left(\sum_{k=1}^N \alpha_{kc} * A_k \right). \quad (7)$$

Similarly to CAM, this map is then upsampled to the input size.

Grad-CAM can be applied to any feature map produced by a convolution, but in practice the last convolutional layer is very often chosen. The authors argue that this layer is “the best compromise between high-level semantics and detailed spatial information” (the latter is lost in fully connected layers, as the feature maps are flattened).

Because of the upsampling step, CAM and Grad-CAM produce maps that are more human-friendly because they contain more connected zones, contrary to other attribution maps obtained with gradient back-propagation that can look very scattered. However, the smallest the feature maps A_k , the blurrier they are, leading to a possible loss of interpretability.

2.3.2 Relevance Back-Propagation

Instead of back-propagating gradients to the level of the input or of the last convolutional layer, Bach et al. [15] proposed to back-propagate the score obtained by a class c , which is called the relevance. This score corresponds to o_c after some post-processing (e.g., softmax), as its value must be positive if class c was identified in the input. At the end of the back-propagation process, the goal is to find the relevance R_u of each feature u of the input (e.g., of each pixel of an image) such that $o_c = \sum_{u \in \mathcal{U}} R_u$.

In their paper, Bach et al. [15] take the example of a fully connected function defined by a matrix of weights w and a bias

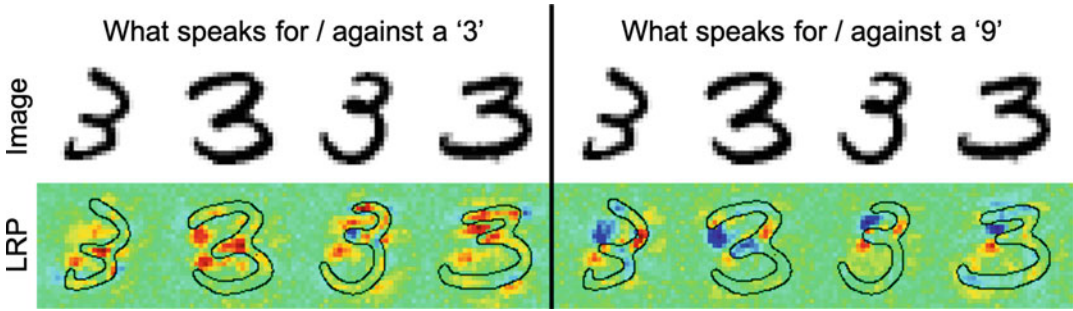


Fig. 9 LRP attribution maps explaining the decision of a neural network trained on MNIST. ©2017 IEEE. (Reprinted, with permission, from [16])

b at layer $l+1$. The value of a node v in feature map $A^{(l+1)}$ is computed during the forward pass by the given formula:

$$A^{(l+1)}(v) = b + \sum_{u \in \mathcal{U}} w_{uv} A^{(l)}(u) \tag{8}$$

During the back-propagation of the relevance, $R^{(l)}(u)$, the value of the relevance at the level of the layer $l+1$ is computed according to the values of the relevance $R^{(l+1)}(v)$ which are distributed according to the weights w learned during the forward pass and the values of $A^{(l)}(v)$:

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}} R^{(l+1)}(v) \frac{A^{(l)}(u) w_{uv}}{\sum_{u' \in \mathcal{U}} A^{(l)}(u') w_{u'v}}. \tag{9}$$

The main issue of the method comes from the fact that the denominator may become (close to) zero, leading to the explosion of the relevance back-propagated. Moreover, it was shown by [11] that when all activations are piece-wise linear (such as ReLU or leaky ReLU), the layer-wise relevance (LRP) method reproduces the output of $\text{gradient} \odot \text{input}$, questioning the usefulness of the method.

This is why Samek et al. [16] proposed two variants of the standard LRP method [15]. Moreover they describe the behavior of the back-propagation in other layers than the linear ones (the convolutional one following the same formula as the linear). They illustrated their method with a neural network trained on MNIST (see Fig. 9). To simplify the equations in the following paragraphs, we now denote the weighted activations as $z_{uv} = A^{(l)}(u) w_{uv}$.

ε-rule The ϵ -rule integrates a parameter $\epsilon > 0$, used to avoid numerical instability. Though it avoids the case of a null denominator, this variant breaks the rule of relevance conservation across layers

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}'} R^{(l+1)}(v) \frac{z_{uv}}{\sum_{u' \in \mathcal{U}} z_{u'v} + \varepsilon \times \text{sign} \left(\sum_{u' \in \mathcal{U}} z_{u'v} \right)}. \quad (10)$$

β -rule The β -rule keeps the conservation of the relevance by treating separately the positive weighted activations z_{uv}^+ from the negative ones z_{uv}^-

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}'} R^{(l+1)}(v) \left((1 + \beta) \frac{z_{uv}^+}{\sum_{u' \in \mathcal{U}} z_{u'v}^+} - \beta \frac{z_{uv}^-}{\sum_{u' \in \mathcal{U}} z_{u'v}^-} \right). \quad (11)$$

Though these two LRP variants improve the numerical stability of the procedure, they imply to choose the values of parameters that may change the patterns in the obtained attribution map.

Deep Taylor Decomposition Deep Taylor decomposition [17] was proposed by the same team as the one that proposed the original LRP method and its variants. It is based on similar principles as LRP: the value of the score obtained by a class c is back-propagated, but the back-propagation rule is based on first-order Taylor expansions.

The back-propagation from node v in at the level of $R^{(l+1)}$ to u at the level of $R^{(l)}$ can be written

$$R^{(l)}(u) = \sum_{v \in \mathcal{V}'} \frac{\partial R^{(l+1)}(v)}{\partial A^{(l)}(u)} \Bigg|_{\tilde{A}_v^{(l)}(u)} \left(A^{(l)}(u) - \tilde{A}_v^{(l)}(u) \right). \quad (12)$$

This rule implies a root point $\tilde{A}_v^{(l)}(u)$ which is close to $A^{(l)}(u)$ and meets a set of constraints depending on v .

2.4 Perturbation Methods

Instead of relying on a backward pass (from the output to the input) as in the previous section, perturbation methods rely on the difference between the value of o_c computed with the original inputs and a locally perturbed input. This process is less abstract for humans than back-propagation methods as we can reproduce it ourselves: if the part of the image that is needed to find the good output is hidden, we are also not able to predict correctly. Moreover, it is model-agnostic and can be applied to any algorithm or deep learning architecture.

The main drawback of these techniques is that the nature of the perturbation is crucial, leading to different attribution maps depending on the perturbation function used. Moreover, Montavon et al. [18] suggest that the perturbation rule should

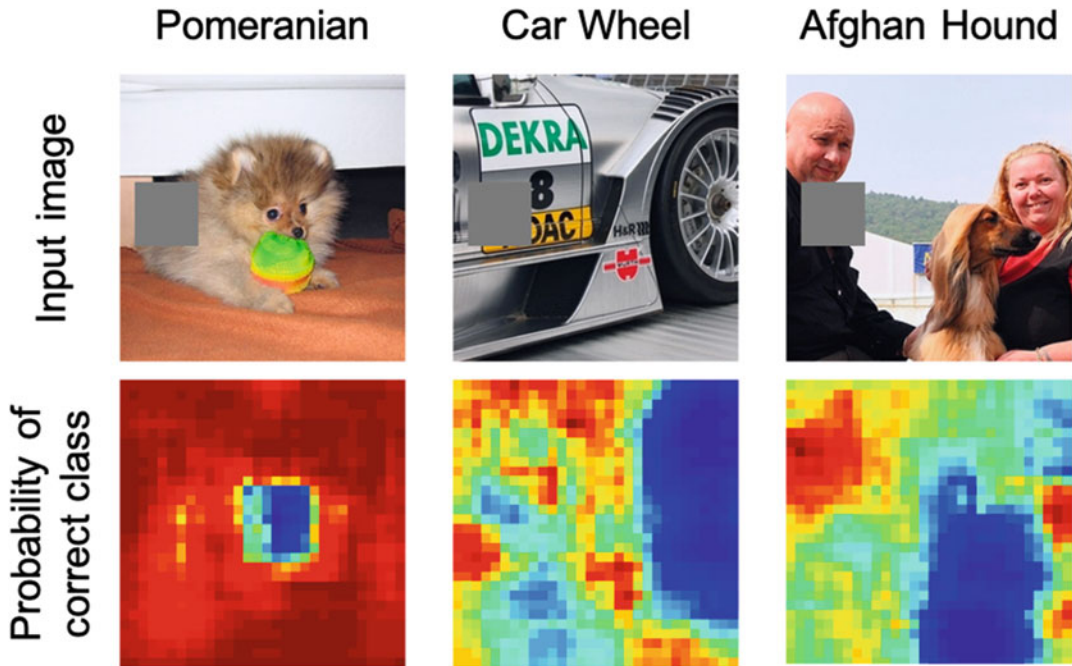


Fig. 10 Attribution maps obtained with standard perturbation. Here the perturbation is a gray patch covering a specific zone of the input as shown in the left column. The attribution maps (second row) display the probability of the true label: the lower the value, the most important it is for the network to correctly identify the label. This kind of perturbation takes the perturbed input out of the training distribution. (Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, ECCV 2014: Visualizing and Understanding Convolutional Networks, [19], 2014)

keep the perturbed input in the training data distribution. Indeed, if it is not the case, one cannot know if the network performance dropped because of the location or the nature of the perturbation.

2.4.1 Standard Perturbation

Zeiler and Fergus [19] proposed the most intuitive method relying on perturbations. This standard perturbation procedure consists in removing information locally in a specific zone of an input X_0 and evaluating if it modifies the output node o_c . The more the perturbation degrades the task performance, the more crucial this zone is for the network to correctly perform the task. To obtain the final attribution map, the input is perturbed according to all possible locations. Examples of attribution maps obtained with this method are displayed in Fig. 10.

As evaluating the impact of the perturbation at each pixel location is computationally expensive, one can choose not to perturb the image at each pixel location but to skip some of them (i.e., scan the image with a stride > 1). This will lead to a smaller attribution map, which needs to be upsampled to be compared to the original input (in the same way as CAM & Grad-CAM).

However, in addition to the problem of the nature of the perturbation previously mentioned, this method presents two drawbacks:

- The attribution maps depend on the size of the perturbation: if the perturbation becomes too large, the perturbation is not local anymore; if it is too small, it is not meaningful anymore (a pixel perturbation cannot cover a pattern).
- Input pixels are considered independently from each other: if the result of a network relies on a combination of pixels that cannot all be covered at the same time by the perturbation, their influence may not be detected.

2.4.2 Optimized Perturbation

To deal with these two issues, Fong and Vedaldi [2] proposed to optimize a perturbation mask covering the whole input. This perturbation mask m has the same size as the input X_0 . Its application is associated with a perturbation function Φ and leads to the computation of the perturbed input X_0^m . Its value at a coordinate u reflects the quantity of information remaining in the perturbed image:

- If $m(u) = 1$, the pixel at location u is not perturbed and has the same value in the perturbed input as in the original input ($X_0^m(u) = X_0(u)$).
- If $m(u) = 0$, the pixel at location u is fully perturbed and the value in the perturbed image is the one given by the perturbation function only ($X_0^m(u) = \Phi(X_0)(u)$).

This principle can be extended to any value between 0 and 1 with the a linear interpolation

$$X_0^m(u) = m(u)X_0(u) + (1 - m(u))\Phi(X_0)(u). \quad (13)$$

Then, the goal is to optimize this mask m according to three criteria:

1. The perturbed input X_0^m should lead to the lowest performance possible.
2. The mask m should perturb the minimum number of pixels possible.
3. The mask m should produce connected zones (i.e., avoid the scattered aspect of gradient maps).

These three criteria are optimized using the following loss:

$$f(X_0^m) + \lambda_1 \|1 - m\|_{\beta_1}^{\beta_1} + \lambda_2 \|\nabla m\|_{\beta_2}^{\beta_2} \quad (14)$$

with f a function that decreases as the performance of the network decreases.

However, the method also presents two drawbacks:



Fig. 11 In this example, the network learned to classify objects in natural images. Instead of masking the maypole at the center of the image, it creates artifacts in the sky to degrade the performance of the network. ©2017 IEEE. (Reprinted, with permission, from [2])

- The values of hyperparameters must be chosen $(\lambda_1, \lambda_2, \beta_1, \beta_2)$ to find a balance between the three optimization criteria of the mask.
- The mask may not highlight the most important pixels of the input but instead create artifacts in the perturbed image to artificially degrade the performance of the network (see Fig. 11).

2.5 Distillation

Approaches described in this section aim at developing a transparent method to reproduce the behavior of a black box one. Then it is possible to consider simple interpretability methods (such as weight visualization) on the transparent method instead of considering the black box.

2.5.1 Local Approximation

LIME Ribeiro et al. [1] proposed local interpretable model-agnostic explanations (LIME). This approach is:

- **Local**, as the explanation is valid in the vicinity of a specific input X_0
- **Interpretable**, as an interpretable model g (linear model, decision tree. . .) is computed to reproduce the behavior of f on X_0
- **Model-agnostic**, as it does not depend on the algorithm trained

This last property comes from the fact that the vicinity of X_0 is explored by sampling variations of X_0 that are perturbed versions of X_0 . Then LIME shares the advantage (model-agnostic) and drawback (perturbation function dependent) of perturbation methods presented in Subheading 2.4. Moreover, the authors specify that, in the case of images, they group pixels of the input in d super-pixels (contiguous patches of similar pixels).

The loss to be minimized to find g specific to the input X_0 is the following:

$$\mathcal{L}(f, g, \pi_{X_0}) + \Omega(g), \tag{15}$$

where π_{X_0} is a function that defines the locality of X_0 (i.e., $\pi_{X_0}(X)$ decreases as X becomes closer to X_0), \mathcal{L} measures how unfaithful g is in approximating f according π_{X_0} , and Ω is a measure of the complexity of g .

Ribeiro et al. [1] limited their search to sparse linear models; however, other assumptions could be made on g .

g is not applied to the input directly but to a binary mask $m \in \{0, 1\}^d$ that transforms the input X in X^m and is applied according to a set of d super-pixels. For each super-pixel u :

1. If $m(u) = 1$, the super-pixel u is not perturbed.
2. If $m(u) = 0$, the super-pixel u is perturbed (i.e., it is grayed).

They used

$$\pi_{X_0}(X) = \exp \frac{(X - X_0)^2}{\sigma^2}$$

and

$$\mathcal{L}(f, g, \pi_{X_0}) = \sum_m \pi_{X_0}(X_0^m) * (f(X_0^m) - g(m))^2.$$

Finally $\Omega(g)$ is the number of non-zero weights of g , and its value is limited to K . This way they select the K super-pixels in X_0 that best explain the algorithm result $f(X_0)$.

SHAP Lundberg and Lee [20] proposed SHAP (SHapley Additive exPlanations), a theoretical framework that encompasses several existing interpretability methods, including LIME. In this framework each of the N features (again, super-pixels for images) is associated with a coefficient ϕ that denotes its contribution to the result. The contribution of each feature is evaluated by perturbing the input X_0 with a binary mask m (see paragraph on LIME). Then the goal is to find an interpretable model g specific to X_0 , such that

$$g(m) = \phi_0 + \sum_1^N \phi_i m_i \quad (16)$$

with ϕ_0 being the output when the input is fully perturbed.

The authors look for an expression of ϕ that respects three properties:

- **Local accuracy** g and f should match in the vicinity of X_0 : $g(m) = f(X_0^m)$.
- **Missingness** Perturbed features should not contribute to the result: $m_i = 0 \rightarrow \phi_i = 0$.
- **Consistency** Let's denote as $m \setminus i$ the mask m in which $m_i = 0$. For any two models f^1 and f^2 , if

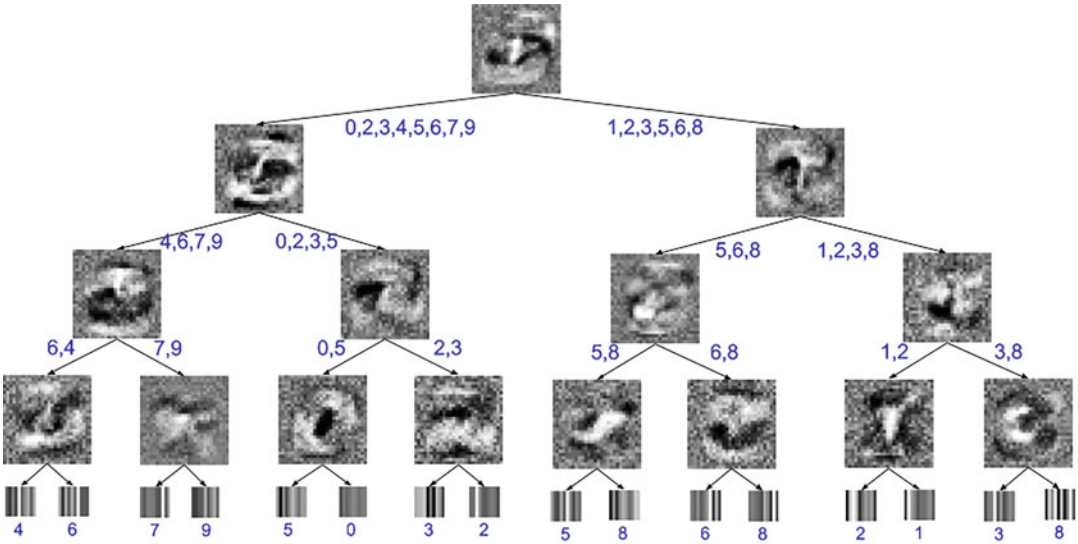


Fig. 12 Visualization of a soft decision tree trained on MNIST. (Adapted from [21]. Permission to reuse was kindly granted by the authors)

$$f^1(X_0^m) - f^1(X_0^{m \setminus i}) \geq f^2(X_0^m) - f^2(X_0^{m \setminus i})$$

then for all $m \in \{0, 1\}^N$ $\phi_i^1 \geq \phi_i^2$ (ϕ^k are the coefficients associated with model f^k).

Lundberg and Lee [20] show that only one expression is possible for the coefficients ϕ , which can be approximated with different algorithms:

$$\phi_i = \sum_{m \in \{0,1\}^N} \frac{|m|!(N - |m| - 1)!}{N!} [f(X_0^m) - f(X_0^{m \setminus i})]. \quad (17)$$

2.5.2 Model Translation

Contrary to local approximation, which provides an explanation according to a specific input X_0 , model translation consists in finding a transparent model that reproduces the behavior of the black box model on the whole data set.

As it was rarely employed in neuroimaging frameworks, this section only discusses the distillation to decision trees proposed in [21] (preprint). For a more extensive review of model translation methods, we refer the reader to [5].

After training a machine learning system f , a binary decision tree g is trained to reproduce its behavior. This tree is trained on a set of inputs X , and each inner node i learns a matrix of weights w_i and biases b_i . The forward pass of X in the node i of the tree is as follows: if $\text{sigmoid}(w_i X + b_i) > 0.5$, then the right leaf node is chosen, else the left leaf node is chosen. After the end of the decision tree’s training, it is possible to visualize at which level which classes were separated to better understand which classes are similar for the

network. It is also possible to visualize the matrices of weights learned by each inner node to identify patterns learned at each class separation. An illustration of this distillation process, on the MNIST data set (hand-written digits), can be found in Fig. 12.

2.6 Intrinsic

Contrary to the previous sections in which interpretability methods could be applied to (almost) any network after the end of the training procedure, the following methods require to design the framework before the training phase, as the interpretability components and the network are trained simultaneously. In the papers presented in this Subheading [22–24], the advantages of these methods are dual: they improve both the interpretability and performance of the network. However, the drawback is that they have to be implemented before training the network, and then they cannot be applied in all cases.

2.6.1 Attention Modules

Attention is a concept in machine learning that consists in producing an attribution map from a feature map and using it to improve learning of another task (such as classification, regression, reconstruction. . .) by making the algorithm focus on the part of the feature map highlighted by the attribution map.

In the deep learning domain, we take as reference [22], in which a network is trained to produce a descriptive caption of natural images. This network is composed of three parts:

1. A convolutional encoder that reduces the dimension of the input image to the size of the feature maps A

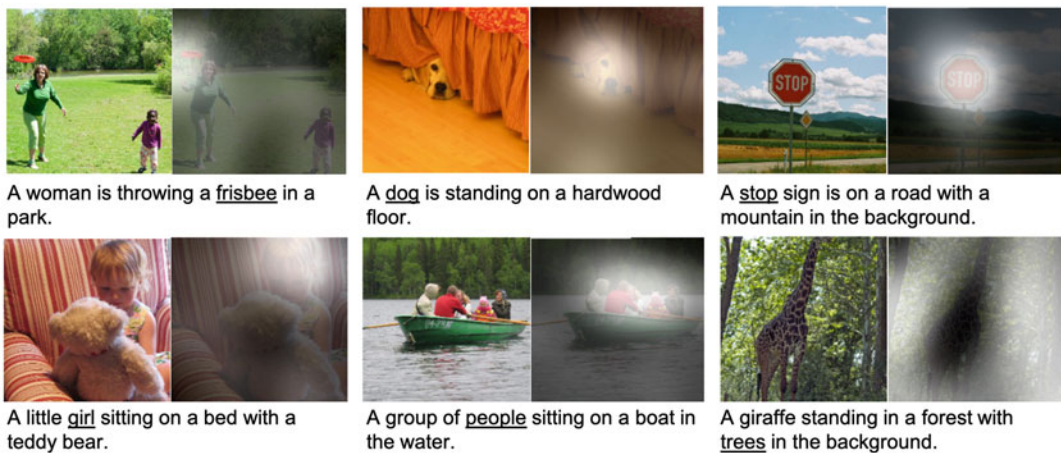


Fig. 13 Examples of images correctly captioned by the network. The focus of the attribution map is highlighted in white and the associated word in the caption is underlined. (Adapted from [22]. Permission to reuse was kindly granted by the authors)

2. An attention module that generates an attribution map S_t from A and the previous hidden state of the long short-term memory (LSTM) network
3. An LSTM decoder that computes the caption from its previous hidden state, the previous word generated, A and S_t

As S_t is of the same size as A (smaller than the input), the result is then upsampled to be overlaid on the input image. As one attribution map is generated per word generated by the LSTM, it is possible to know where the network focused when generating each word of the caption (*see* Fig. 13). In this example, the attribution map is given to a LSTM, which uses it to generate a context vector z_t by applying a function ϕ to A and S_t .

More generally in CNNs, the point-wise product of the attribution map S and the feature map A is used to generate the refined feature map A' which is given to the next layers of the network. Adding an attention module implies to make new choices for the architecture of the model: its location (on lower or higher feature maps) may impact the performance of the network. Moreover, it is possible to stack several attention modules along the network, as it was done in [23].

2.6.2 Modular Transparency

Contrary to the studies of the previous sections, the frameworks of these categories are composed of several networks (modules) that interact with each other. Each module is a black box, but the transparency of the function, or the nature of the interaction

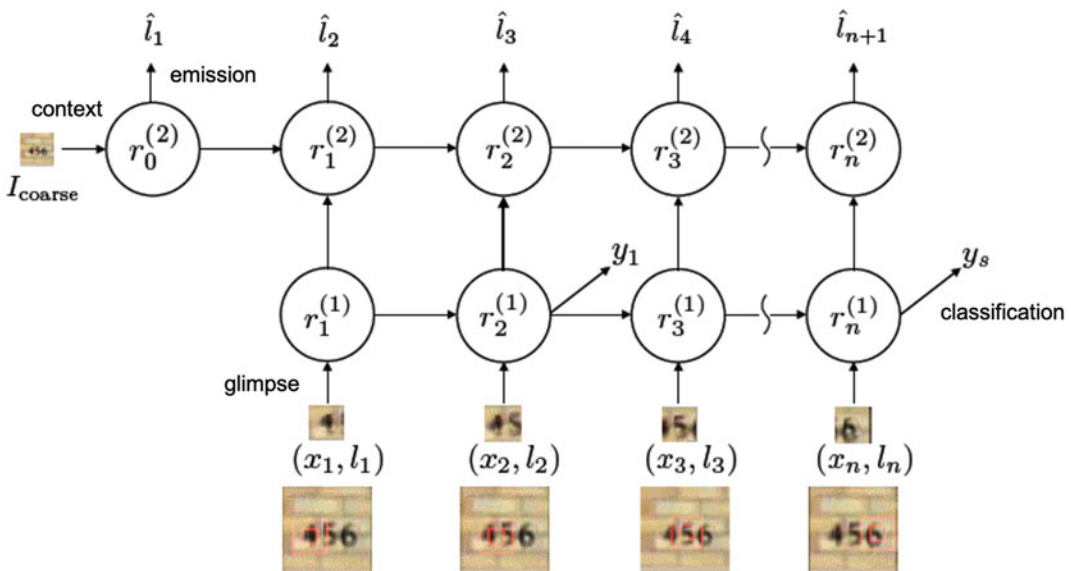


Fig. 14 Framework with modular transparency browsing an image to compute the output at the global scale. (Adapted from [24]. Permission to reuse was kindly granted by the authors)

between them, allows understanding how the system works globally and extracting interpretability metrics from it.

A large variety of setups can be designed following this principle, and it is not possible to draw a more detailed general rule for this section. We will take the example described in [24], which was adapted to neuroimaging data (*see* Subheading 3.6), to illustrate this section, though it may not be representative of all the aspects of modular transparency.

Ba et al. [24] proposed a framework (illustrated in Fig. 14) to perform the analysis of an image in the same way as a human, by looking at successive relevant locations in the image. To perform this task, they assemble a set of networks that interact together:

- **Glimpse network** This network takes as input a patch of the input image and the location of its center to output a context vector that will be processed by the recurrent network. Then this vector conveys information on the main features in a patch and its location.
- **Recurrent network** This network takes as input the successive context vectors and update its hidden state that will be used to find the next location to look at and to perform the learned task at the global scale (in the original paper a classification of the whole input image).
- **Emission network** This network takes as input the current state of the recurrent network and outputs the next location to look at. This will allow computing the patch that will feed the glimpse network.
- **Context network** This network takes as input the whole input at the beginning of the task and outputs the first context vector to initialize the recurrent network.
- **Classification network** This network takes as input the current state of the recurrent network and outputs a prediction for the class label.

The global framework can be seen as interpretable as it is possible to review the successive processed locations.

2.7 Interpretability Metrics

To evaluate the reliability of the methods presented in the previous sections, one cannot only rely on qualitative evaluation. This is why interpretability metrics that evaluate attribution maps were proposed. These metrics may evaluate different properties of attribution maps.

- **Fidelity** evaluates if the zones highlighted by the map influence the decision of the network.
- **Sensitivity** evaluates how the attribution map changes according to small changes in the input X_0 .

- **Continuity** evaluates if two close data points lead to similar attribution maps.

In the following, Γ is an interpretability method computing an attribution map S of the black box network f and an input X_0 .

2.7.1 *(In)fidelity*

Yeh et al. [25] proposed a measure of infidelity of Γ based on perturbations applied according to a vector m of the same shape as the attribution map S . The explanation is infidel if perturbations applied in zones highlighted by S on X_0 lead to negligible changes in $f(X_0^m)$ or, on the contrary, if perturbations applied in zones not highlighted by S on X_0 lead to significant changes in $f(X_0^m)$. The associated formula is

$$\text{INFD}(\Gamma, f, X_0) = \mathbb{E}_m \left[\sum_i \sum_j m_{ij} \Gamma(f, X_0)_{ij} - (f(X_0) - f(X_0^m))^2 \right]. \quad (18)$$

2.7.2 *Sensitivity*

Yeh et al. [25] also gave a measure of sensitivity. As suggested by the definition, it relies on the construction of attribution maps according to inputs similar to X_0 : \tilde{X}_0 . As changes are small, sensitivity depends on a scalar ϵ set by the user, which corresponds to the maximum difference allowed between X_0 and \tilde{X}_0 . Then sensitivity corresponds to the following formula:

$$\text{SENS}_{\max}(\Gamma, f, X_0, \epsilon) = \max_{\|\tilde{X}_0 - X_0\| \leq \epsilon} \|\Gamma(f, \tilde{X}_0) - \Gamma(f, X_0)\|. \quad (19)$$

2.7.3 *Continuity*

Continuity is very similar to sensitivity, except that it compares different data points belonging to the input domain \mathcal{X} , whereas sensitivity may generate similar inputs with a perturbation method. This measure was introduced in [18] and can be computed using the following formula:

$$\text{CONT}(\Gamma, f, \mathcal{X}) = \max_{X_1, X_2 \in \mathcal{X} \ \& \ X_1 \neq X_2} \frac{\|\Gamma(f, X_1) - \Gamma(f, X_2)\|_1}{\|X_1 - X_2\|_2}. \quad (20)$$

As these metrics rely on perturbation, they are also influenced by the nature of the perturbation and may lead to different results, which is a major issue (*see* Subheading 4). Other metrics were also proposed and depend on the task learned by the network: for example, in the case of a classification, statistical tests can be conducted between attribution maps of different classes to assess whether they differ according to the class they explain.

3 Application of Interpretability Methods to Neuroimaging Data

In this section, we provide a non-exhaustive review of applications of interpretability methods to neuroimaging data. In most cases, the focus of articles is prediction/classification rather than the interpretability method, which is just seen as a tool to analyze the results. Thus, authors do not usually motivate their choice of an interpretability method. Another key consideration here is the spatial registration of brain images, which enables having brain regions roughly at the same position between subjects. This technique is of paramount importance as attribution maps computed for registered images can then be averaged or used to automatically determine the most important brain areas, which would not be possible with unaligned images. All the studies presented in this section are summarized in Table 3.

Table 3
Summary of the studies applying interpretability methods to neuroimaging data which are presented in Subheading 3

Study	Data set	Modality	Task	Interpretability method	Section
Abrol et al. [28]	ADNI	T1w	AD classification	FM visualization, Perturbation	3.2, 3.4
Bae et al. [32]	ADNI	sMRI	AD classification	Perturbation	3.4
Ball et al. [33]	PING	T1w	Age prediction	Weight visualization, SHAP	3.1, 3.5
Biffi et al. [29]	ADNI	T1w	AD classification	FM visualization	3.2
Böhle et al. [34]	ADNI	T1w	AD classification	LRP, Guided back-propagation	3.3
Burduja et al. [35]	RSNA	CT scan	Intracranial Hemorrhage detection	Grad-CAM	3.3
Cecotti and Gräser [26]	in-house	EEG	P300 signals detection	Weight visualization	3.1
Dyrba et al. [36]	ADNI	T1w	AD classification	DeconvNet, Deep Taylor decomposition, Gradient \odot Input, LRP, Grad-CAM	3.3
Eitel and Ritter [37]	ADNI	T1w	AD classification	Gradient \odot Input, Guided back-propagation, LRP, Perturbation	3.3, 3.4

(continued)

Table 3
(continued)

Study	Data set	Modality	Task	Interpretability method	Section
Eitel et al. [38]	ADNI, in-house	T1w	Multiple Sclerosis detection	Gradient \odot Input, LRP	3.3
Fu et al. [39]	CQ500, RSNA	CT scan	Detection of Critical Findings in Head CT scan	Attention mechanism	3.6
Gutiérrez-Becker and Wachinger [40]	ADNI	T1w	AD classification	Perturbation	3.4
Hu et al. [41]	ADNI, NIFD	T1w	AD/CN/FTD classification	Guided back-propagation	3.3
Jin et al. [42]	ADNI, in-house	T1w	AD classification	Attention mechanism	3.6
Lee et al. [43]	ADNI	T1w	AD classification	Modular transparency	3.6
Leming et al. [31]	OpenfMRI, ADNI, ABIDE, ABIDE II, ABCD, NDAR, ICBM, UK Biobank, 1000FC	fMRI	Autism classification Sex classification Task vs rest classification	FM visualization, Grad-CAM	3.2, 3.3
Magesh et al. [44]	PPMI	SPECT	Parkinson’s disease detection	LIME	3.5
Martinez-Murcia et al. [30]	ADNI	T1w	AD classification Prediction of neuropsychological tests & other clinical variables	FM visualization	3.2
Nigri et al. [45]	ADNI, AIBL	T1w	AD classification	Perturbation, Swap test	3.4
Oh et al. [27]	ADNI	T1w	AD classification	FM visualization, Standard back-propagation, Perturbation	3.2, 3.3, 3.4
Qiu et al. [46]	ADNI, AIBL, FHS, NACC	T1w	AD classification	Modular transparency	3.6
Ravi et al. [47]	ADNI	T1w	CN/MCI/AD reconstruction	Modular transparency	3.6

(continued)

Table 3
(continued)

Study	Data set	Modality	Task	Interpretability method	Section
Rieke et al. [48]	ADNI	T1w	AD classification	Standard back-propagation, Guided back-propagation, Perturbation, Brain area occlusion	3.3, 3.4
Tang et al. [49]	UCD-ADC, Brain Bank	Histology	Detection of amyloid- β pathology	Guided back-propagation, Perturbation	3.3, 3.4
Wood et al. [50]	ADNI	T1w	AD classification	Modular transparency	3.6

Data sets: 1000FC, 1000 Functional Connectomes; ABCD, Adolescent Brain Cognitive Development; ABIDE, Autism Brain Imaging Data Exchange; ADNI, Alzheimer’s Disease Neuroimaging Initiative; AIBL, Australian Imaging, Biomarkers and Lifestyle; FHS, Framingham Heart Study; ICBM, International Consortium for Brain Mapping; NACC, National Alzheimer’s Coordinating Center; NDAR, National Database for Autism Research; NIFD, frontotemporal lobar degeneration neuroimaging initiative; PING, Pediatric Imaging, Neurocognition and Genetics; PPMI, Parkinson’s Progression Markers Initiative; RSNA, Radiological Society of North America 2019 Brain CT Hemorrhage data set; UCD-ADC Brain Bank, University of California Davis Alzheimer’s Disease Center Brain Bank

Modalities: CT, computed tomography; EEG, electroencephalography; fMRI, functional magnetic resonance imaging; SMRI, structural magnetic resonance imaging; SPECT, single-photon emission computed tomography; T1w, T1-weighted [magnetic resonance imaging]

Tasks: AD, Alzheimer’s disease; CN, cognitively normal; FTD, frontotemporal dementia; MCI, mild cognitive impairment

Interpretability methods: FM, feature maps; Grad-CAM, gradient-weighted class activation mapping; LIME, local interpretable model-agnostic explanations; LRP, layer-wise relevance; SHAP, SHapley Additive exPlanations

This section ends with the presentation of benchmarks conducted in the literature to compare different interpretability methods in the context of brain disorders.

3.1 Weight Visualization Applied to Neuroimaging

As the focus of this chapter is on non-transparent models, such as deep learning ones, weight visualization was only rarely found. However, this was the method chosen by Cecotti and Gräser [26], who developed a CNN architecture adapted to weight visualization to detect P300 signals in electroencephalograms (EEG). The input of this network is a matrix with rows corresponding to the 64 electrodes and columns to 78 time points. The two first layers of the networks are convolutions with rectangular filters: the first filters (size 1×64) combine the electrodes, whereas the second ones (13×1) find time patterns. Then, it is possible to retrieve a coefficient per electrode by summing the weights associated with this electrode across the different filters and to visualize the results in the electroencephalogram space as shown in Fig. 15.

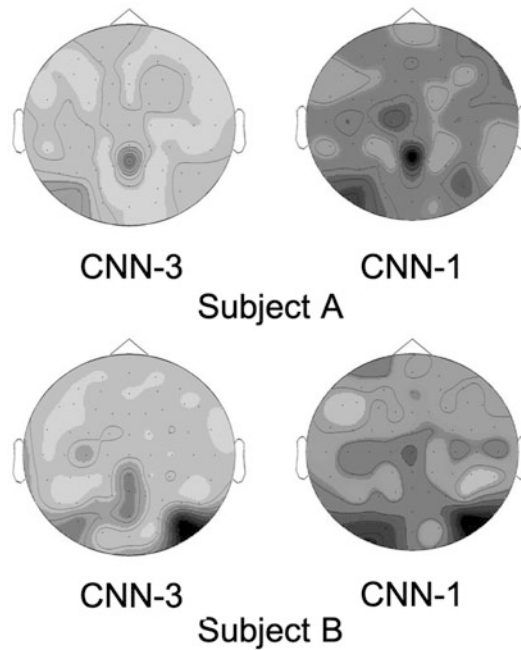


Fig. 15 Relative importance of the electrodes for signal detection in EEG using two different architectures (CNN-1 and CNN-3) and two subjects (A and B) using CNN weight visualization. Dark values correspond to weights with a high absolute value while white values correspond to weights close to 0. ©2011 IEEE. (Reprinted, with permission, from [26])

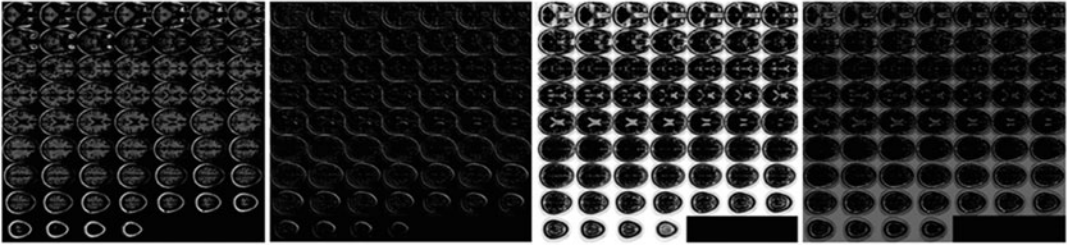
3.2 Feature Map Visualization Applied to Neuroimaging

Contrary to the limited application of weight visualization, there is an extensive literature about leveraging individual feature maps and latent spaces to better understand how models work. This goes from the visualization of these maps or their projections [27–29], to the analysis of neuron behavior [30, 31], through sampling in latent spaces [29].

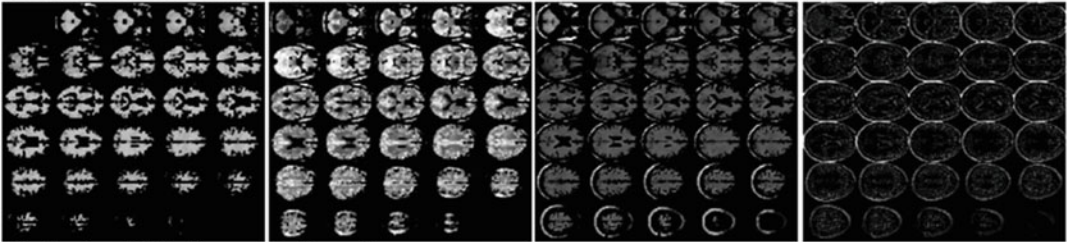
Oh et al. [27] displayed the feature maps associated with the convolutional layers of CNNs trained for various Alzheimer’s disease status classification tasks (Fig. 16). In the first two layers, the extracted features were similar to white matter, cerebrospinal fluid, and skull segmentations, while the last layer showcased sparse, global, and nearly binary patterns. They used this example to emphasize the advantage of using CNNs to extract very abstract and complex features rather than using custom algorithms for feature extraction [27].

Another way to visualize a feature map is to project it in a two- or three-dimensional space to understand how it is positioned with respect to other feature maps. Abrol et al. [28] projected the features obtained after the first dense layer of a ResNet architecture onto a two-dimensional space using the classical t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction technique. For the classification task of Alzheimer’s disease

Outputs of first convolutional layers



Outputs of second convolutional layers



Outputs of third convolutional layers

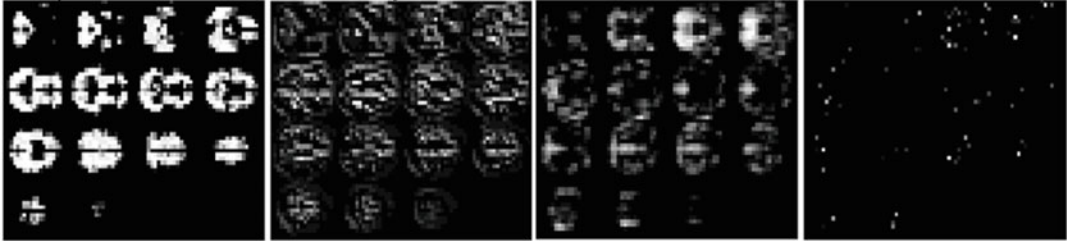


Fig. 16 Representation of a selection of feature maps (outputs of 4 filters on 10 for each layer) obtained for a single individual. (Adapted from [27] (CC BY 4.0))

statuses, they observed that the projections were correctly ordered according to the disease severity, supporting the correctness of the model [28]. They partitioned these projections into three groups: Far-AD (more extreme Alzheimer’s Disease patients), Far-CN (more extreme Cognitively Normal participants), and Fused (a set of images at the intersection of AD and CN groups). Using a t-test, they were able to detect and highlight voxels presenting significant differences between groups (Fig. 17).

Biffi et al. [29] not only used feature map visualization but also sampled the feature space. Indeed, they trained a ladder variational autoencoder framework to learn hierarchical latent representations of 3D hippocampal segmentations of control subjects and Alzheimer’s disease patients. A multilayer perceptron was jointly trained on top of the highest two-dimensional latent space to classify anatomical shapes. While lower spaces needed a dimensionality reduction technique (i.e., t-SNE), the highest latent space could directly be visualized, as well as the anatomical variability it captured in the initial input space, by leveraging the generative process of the model. This sampling enabled an easy visualization and quantification of the anatomical differences between each class.

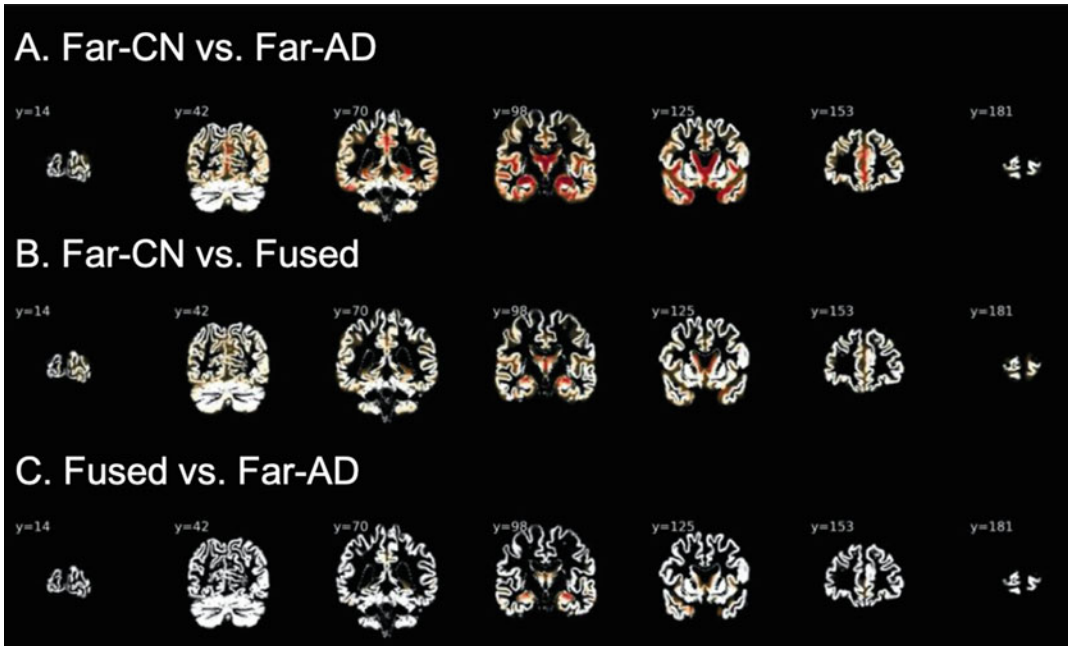


Fig. 17 Difference in neuroimaging space between groups defined thanks to t-SNE projection. Voxels showing significant differences post false discovery rate (FDR) correction ($p < 0.05$) are highlighted. (Reprinted from *Journal of Neuroscience Methods*, 339, [28], 2020, with permission from Elsevier)

Finally, it may be very informative to better understand the behavior of neurons and what they are encoding. After training deep convolutional autoencoders to reconstruct MR images, segmented gray matter maps, and white matter maps, Martinez-Murcia et al. [30] computed correlations between each individual hidden neuron value and clinical information (e.g., age, mini-mental state examination) which allowed them to determine to which extent this information was encoded in the latent space. This way they determined which clinical data was the most strongly associated. Using a collection of nine different MRI data sets, Leming et al. [31] trained CNNs for various classification tasks (autism vs typically developing, male vs female, and task vs rest). They computed a diversity coefficient for each filter of the second layer based on its output feature map. They counted how many different data sets maximally activated each value of this feature map: if they were mainly activated by one source of data, the coefficient would be close to 0, whereas if they were activated by all data sets, it would be close to 1. This allows assessing the layer stratification, i.e., to understand if a given filter was mostly maximally activated by one phenotype or by a diverse population. They found out that a few filters were only maximally activated by images from a single MRI data set and that the diversity coefficient was not normally distributed across filters, having generally two peaks at the

beginning and at the end of the spectrum, respectively, exhibiting the stratification and strongly diverse distribution of the filters.

3.3 Back-Propagation Methods Applied to Neuroimaging

Back-propagation methods are the most popular methods to interpret models, and a wide range of these algorithms have been used to study brain disorders: standard and guided back-propagation [27, 34, 37, 41, 48], gradient \odot input [36–38], Grad-CAM [35, 36], guided Grad-CAM [49], LRP [34, 36–38], DeconvNet [36], and deep Taylor Decomposition [36].

3.3.1 Single Interpretation

Some studies implemented a single back-propagation method and exploited it to find which brain regions are exploited by their algorithm [27, 31, 41], to validate interpretability methods [38], or to provide attribution maps to physicians to improve clinical guidance [35].

Oh et al. [27] used the standard back-propagation method to interpret CNNs for classification of Alzheimer’s disease statuses. They showed that the attribution maps associated with the prediction of the conversion of prodromal patients to dementia included more complex representations, less focused on the hippocampi, than the ones associated with classification between demented patients from cognitively normal participants (*see* Fig. 18). In the context of autism, Leming et al. [31] used the Grad-CAM

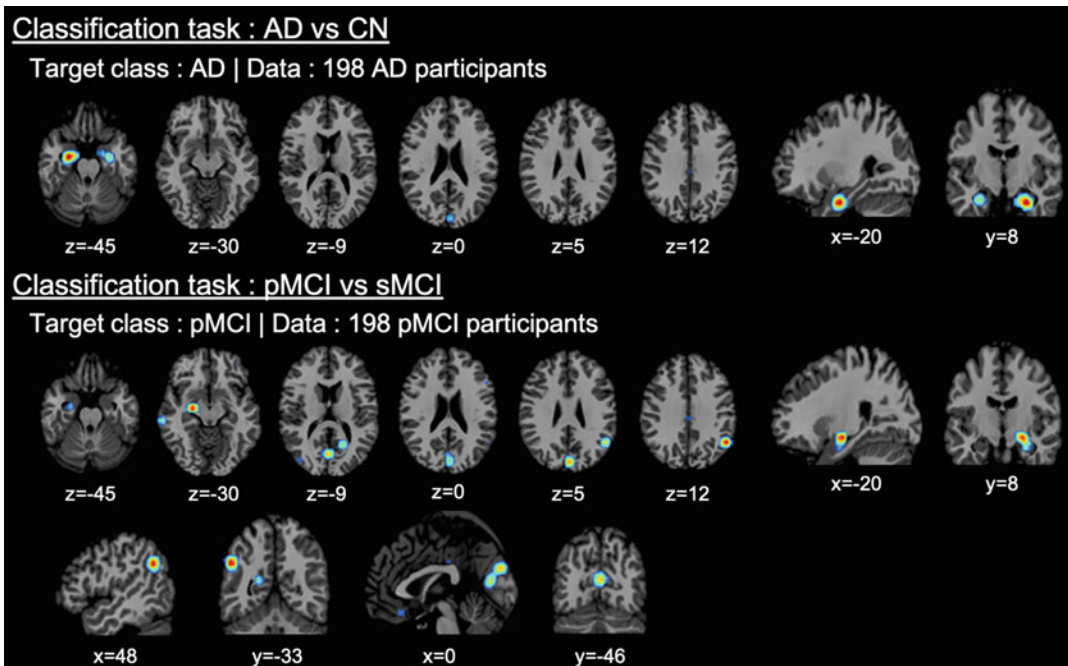


Fig. 18 Distribution of discriminant regions obtained with gradient back-propagation in the classification of demented patients and cognitively normal participants (top part, AD vs CN) and the classification of stable and progressive mild cognitive impairment (bottom part, sMCI vs pMCI). (Adapted from [27] (CC BY 4.0))

algorithm to determine the most important brain connections from functional connectivity matrices. However, the authors pointed out that without further work, this visualization method did not allow understanding the underlying reason of the attribution of a given feature: for instance, one cannot know if a set of edges is important because it is under-connected or over-connected. Finally, Hu et al. [41] used attribution maps produced by guided back-propagation to quantify the difference in the regions used by their network to characterize Alzheimer's disease or frontotemporal dementia.

The goal of Eitel et al. [38] was different. Instead of identifying brain regions related to the classification task, they exhibited with LRP that transfer learning between networks trained on different diseases (Alzheimer's disease to multiple sclerosis) and different MRI sequences enabled obtaining attribution maps focused on a smaller number of lesion areas. However, the authors pointed out that it would be necessary to confirm their results on larger data sets.

Finally, Burduja et al. [35] trained a CNN-LSTM model to detect various hemorrhages from brain computed tomography (CT) scans. For each positive slice coming from controversial or difficult scans, they generated Grad-CAM-based attribution maps and asked a group of radiologists to classify them as correct, partially correct, or incorrect. This classification allowed them to determine patterns for each class of maps and better understand which characteristics radiologists expected from these maps to be considered as correct and thus useful in practice. In particular, radiologists described maps including any type of hemorrhage as incorrect as soon as some of the hemorrhages were not highlighted, while the model only needed to detect one hemorrhage to correctly classify the slice as pathological.

3.3.2 Comparison of Several Interpretability Methods

Papers described in this section used several interpretability methods and compared them in their particular context. However, as the benchmark of interpretability methods is the focus of Subheading 4.3, which also include other types of interpretability than back-propagation, we will only focus here on what conclusions were drawn from the attribution maps.

Dyrba et al. [36] compared DeconvNet, guided back-propagation, deep Taylor decomposition, gradient \odot input, LRP (with various rules), and Grad-CAM methods for classification of Alzheimer's disease, mild cognitive impairment, and normal cognition statuses. In accordance with the literature, they obtained a highest attention given to the hippocampus for both prodromal and demented patients.

Böhle et al. [34] compared two methods, LRP with β -rule and guided back-propagation for Alzheimer's disease status classification. They found that LRP attribution maps highlight the individual differences between patients and then that they could be used as a tool for clinical guidance.

3.4 Perturbation Methods Applied to Neuroimaging

The standard perturbation method has been widely used in the study of Alzheimer's disease [32, 37, 45, 48] and related symptoms (amyloid- β pathology) [49]. However, most of the time, authors do not train their model with perturbed images. Hence, to generate explanation maps, the perturbation method uses images outside the distribution of the training set, which may call into question the relevance of the predictions and thus the reliability of attention maps.

3.4.1 Variants of the Perturbation Method Tailored to Neuroimaging

Several variations of the perturbation method have been developed to adapt to neuroimaging data. The most common variation in brain imaging is the brain area perturbation method, which consists in perturbing entire brain regions according to a given brain atlas, as done in [27, 28, 48]. In their study of Alzheimer's disease, Abrol et al. [28] obtained high values in their attribution maps for the usually discriminant brain regions, such as the hippocampus, the amygdala, the inferior and superior temporal gyri, and the fusiform gyrus. Rieke et al. [48] also obtained results in accordance with the medical literature and noted that the brain area perturbation method led to a less scattered attribution map than the standard method (Fig. 19). Oh et al. [27] used the method to compare the attribution maps of two different tasks: (1) demented patients vs cognitively normal participants and (2) stable vs progressive mild cognitively impaired patients and noted that the regions targeted

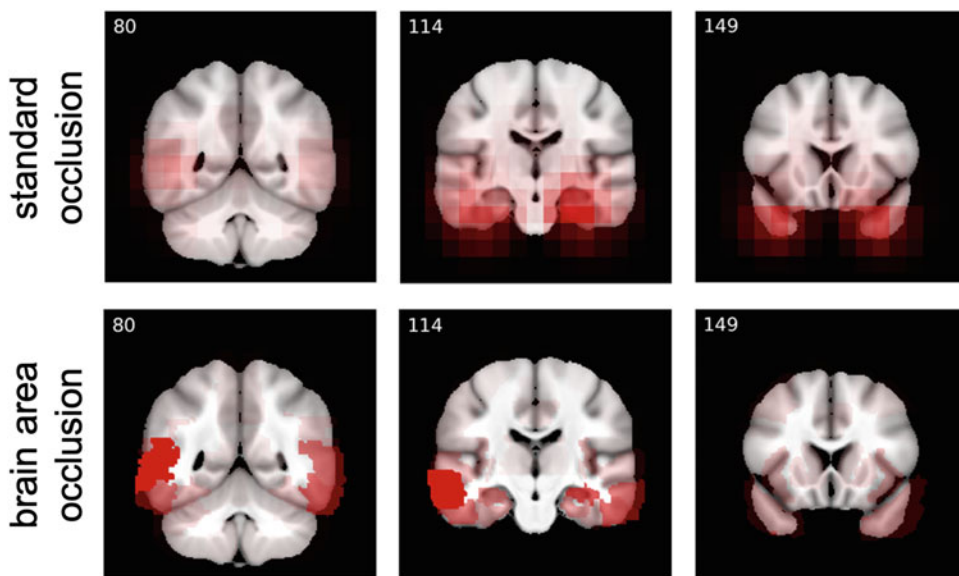


Fig. 19 Mean attribution maps obtained on demented patients. The first row corresponds to the standard and the second one to the brain area perturbation method. (Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, MLCN 2018, DLF 2018, IMIMIC 2018: Understanding and Interpreting Machine Learning in Medical Image Computing Applications, [48], 2018)

for the first task were shared with the second one (medial temporal lobe) but that some regions were specific to the second task (parts of the parietal lobe).

Gutiérrez-Becker and Wachinger [40] adapted the standard perturbation method to a network that classified clouds of points extracted from neuroanatomical shapes of brain regions (e.g., left hippocampus) between different states of Alzheimer’s disease. For the perturbation step, the authors set to 0 the coordinates of a given point x and the ones of its neighbors to then assess the relevance of the point x . This method allows easily generating and visualizing a 3D attribution map of the shapes under study.

3.4.2 Advanced Perturbation Methods

More advanced perturbation-based methods have also been used in the literature. Nigri et al. [45] compared a classical perturbation method to a swap test. The swap test replaces the classical perturbation step by a swapping step where patches are exchanged between the input brain image and a reference image chosen according to the model prediction. This exchange is possible as brain images were registered and thus brain regions are positioned in roughly the same location in each image.

Finally, Thibeau-Sutre et al. [51] used the optimized version of the perturbation method to assess the robustness of CNNs in identifying regions of interest for Alzheimer’s disease detection. They applied optimized perturbations on gray matter maps extracted from T1w MR images, and the perturbation method consisted in increasing the value of the voxels to transform patients into controls. This process aimed at stimulating gray matter reconstruction to identify the most important regions that needed to be “de-atrophied” to be considered again as normal. However, they unveiled a lack of robustness of the CNN: different retrains led to different attribution maps (shown in Fig. 20) even though the performance did not change.

3.5 Distillation Methods Applied to Neuroimaging

Distillation methods are less commonly used, but some very interesting use cases can be found in the literature on brain disorders, with methods such as LIME [44] or SHAP [33].

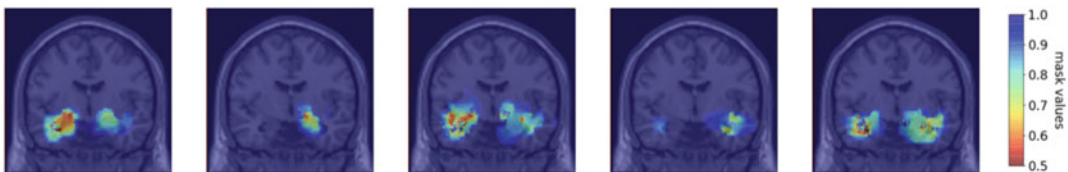


Fig. 20 Coronal view of the mean attribution masks on demented patients obtained for five reruns of the same network with the optimized perturbation method. (Adapted with permission from Medical Imaging 2020: Image Processing, [51].)

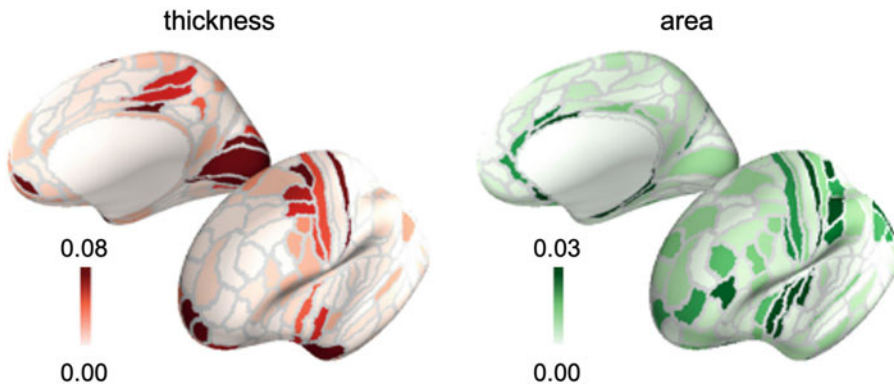


Fig. 21 Mean absolute feature importance (SHAP values) averaged across all subjects for XGBoost on regional thicknesses (red) and areas (green). (Adapted from [33] (CC BY 4.0))

Magesh et al. [44] used LIME to interpret a CNN for Parkinson’s disease detection from single-photon emission computed tomography (SPECT) scans. Most of the time, the most relevant regions are the putamen and the caudate (which is clinically relevant), and some patients also showed an anomalous increase in dopamine activity in nearby areas, which is a characteristic feature of late-stage Parkinson’s disease. The authors did not specify how they extracted the “super-pixels” necessary to the application of the method, though it could have been interesting to consider neuro-anatomical regions instead of creating the voxel groups with an agnostic method.

Ball et al. [33] used SHAP to obtain explanations at the individual level from three different models trained to predict participants’ age from regional cortical thicknesses and areas: regularized linear model, Gaussian process regression, and XGBoost (Fig. 21). The authors exhibited a set of regions driving predictions for all models and showed that regional attention was highly correlated on average with weights of the regularized linear model. However, they showed that while being consistent across models and training folds, explanations of SHAP at the individual level were generally not correlated with feature importance obtained from the weight analysis of the regularized linear model. The authors also exemplified that the global contribution of a region to the final prediction error (“brain age delta”), even with a high SHAP value, was in general small, which indicated that this error was best explained by changes spread across several regions [33].

3.6 Intrinsic Methods Applied to Neuroimaging

3.6.1 Attention Modules

Attention modules have been increasingly used in the past couple of years, as they often allow a boost in performance while being rather easy to implement and interpret. To diagnose various brain diseases from brain CT images, Fu et al. [39] built a model integrating a “two-step attention” mechanism that selects both the most

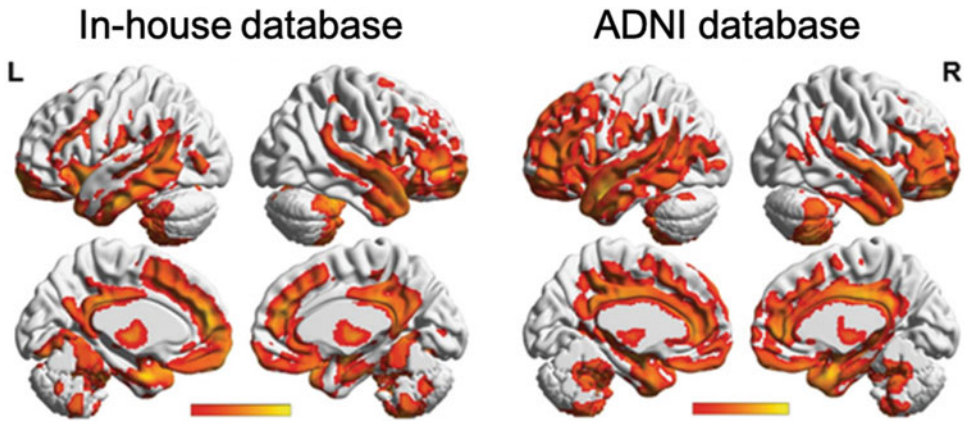


Fig. 22 Attribution maps (left, in-house database; right, ADNI database) generated by an attention mechanism module, indicating the discriminant power of various brain regions for Alzheimer's disease diagnosis. (Adapted from [42] (CC BY 4.0))

important slices and the most important pixels in each slice. The authors then leveraged these attention modules to retrieve the five most suspicious slices and highlight the areas with the more significant attention.

In their study of Alzheimer's disease, Jin et al. [42] used a 3D attention module to capture the most discriminant brain regions used for Alzheimer's disease diagnosis. As shown in Fig. 22, they obtained significant correlations between attention patterns for two independent databases. They also obtained significant correlations between regional attention scores of two different databases, which indicated a strong reproducibility of the results.

3.6.2 Modular Transparency

Modular transparency has often been used in brain imaging analysis. A possible practice consists in first generating a target probability map of a black box model, before feeding this map to a classifier to generate a final prediction, as done in [43, 46].

Qiu et al. [46] used a convolutional network to generate an attribution map from patches of the brain, highlighting brain regions associated with Alzheimer's disease diagnosis (see Fig. 23). Lee et al. [43] first parcellated gray matter density maps into 93 regions. For each of these regions, several deep neural networks were trained on randomly selected voxels, and their outputs were averaged to obtain a mean regional disease probability. Then, by concatenating these regional probabilities, they generated a region-wise disease probability map of the brain, which was further used to perform Alzheimer's disease detection.

The approach of Ba et al. [24] was also applied to Alzheimer's disease detection [50] (preprint). Though that work is still a preprint, the idea is interesting as it aims at reproducing the way a radiologist looks at an MR image. The main difference with [24] is

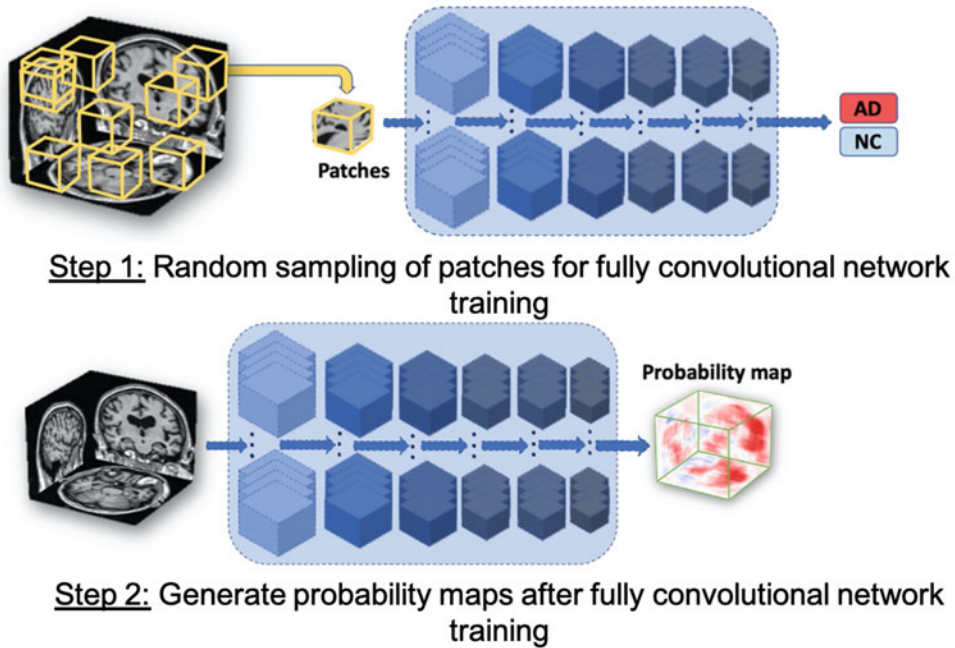


Fig. 23 Randomly selected samples of T1-weighted full MRI volumes are used as input to learn the Alzheimer's disease status at the individual level (Step 1). The application of the model to whole images leads to the generation of participant-specific disease probability maps of the brain (Step 2). (Adapted from *Brain: A Journal of Neurology*, 143, [46], 2020, with permission of Oxford University Press)

the initialization, as the context network does not take as input the whole image but clinical data of the participant. Then the framework browses the image in the same way as in the original paper: a patch is processed by a recurrent neural network and from its internal state the glimpse network learns which patch should be looked at next. After a fixed number of iterations, the internal state of the recurrent neural network is processed by a classification network that gives the final outcome. The whole system is interpretable as the trajectory of the locations (illustrated in Fig. 24) processed by the framework allows understanding which regions are more important for the diagnosis. However, this framework may have a high dependency to clinical data: as the initialization depends on scores used to diagnose Alzheimer's disease, the classification network may learn to classify based on the initialization only, and most of the trajectory may be negligible to assess the correct label.

Another framework, the DaniNet, proposed by Ravi et al. [47], is composed of multiple networks, each with a defined function, as illustrated in Fig. 25.

- The conditional deep autoencoder (in orange) learns to reduce the size of the slice x to a latent variable Z (encoder part) and

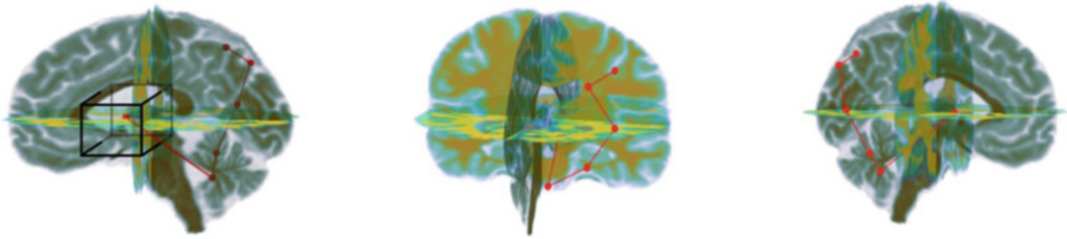


Fig. 24 Trajectory taken by the framework for a participant from the ADNI test set. A bounding box around the first location attended to is included to indicate the approximate size of the glimpse that the recurrent neural network receives; this is the same for all subsequent locations. (Adapted from [50]. Permission to reuse was kindly granted by the authors)

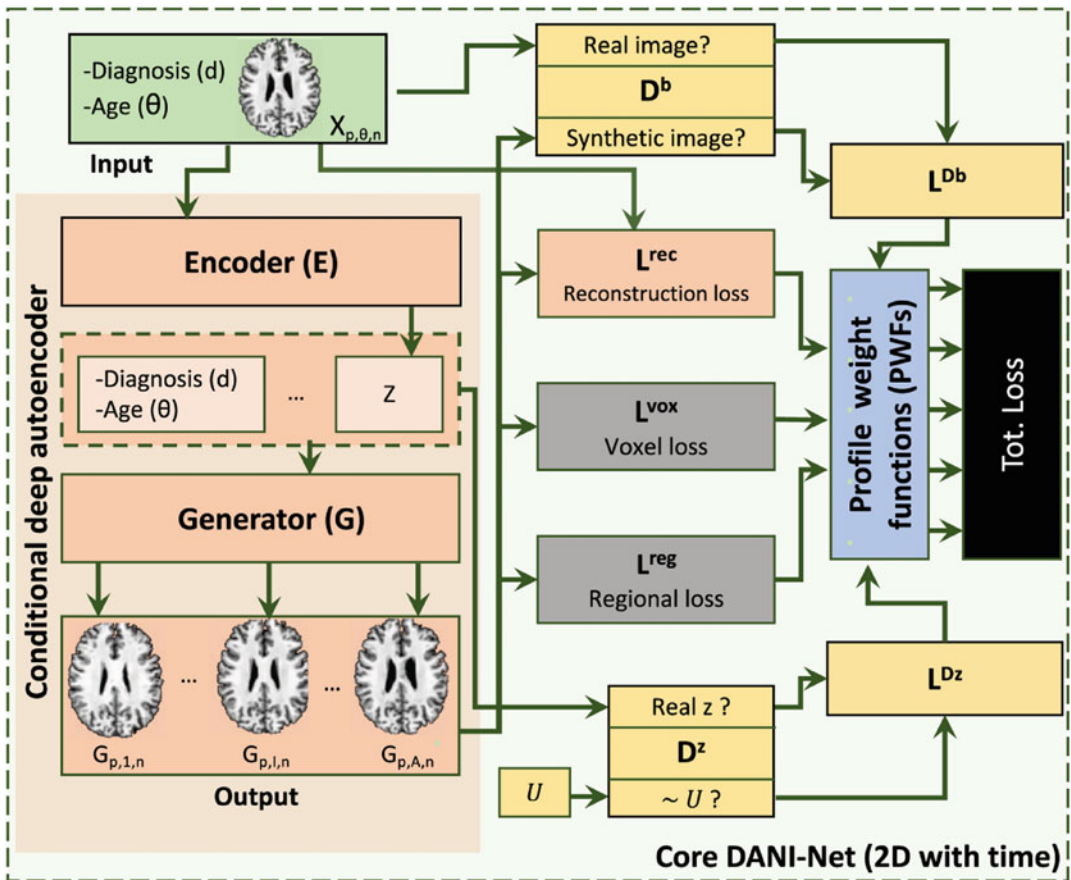


Fig. 25 Pipeline used for training the proposed DaniNet framework that aims to learn a longitudinal model of the progression of Alzheimer’s disease. (Adapted from [47] (CC BY 4.0))

then to reconstruct the original image based on Z and two additional variables: the diagnosis and age (generator part). Its performance is evaluated thanks to the reconstruction loss L^{rec} .

- Discriminator networks (in yellow) either force the encoder to take temporal progression into account (D_z) or try to determine if the output of the generator are real or generated images (D_b).
- Biological constraints (in grey) force the previous generated image of the same participant to be less atrophied than the next one (voxel loss) and learn to find the diagnosis thanks to regions of the generated images (regional loss).
- Profile weight functions (in blue) aim at finding appropriate weights for each loss to compute the total loss.

The assembly of all these components allows learning a longitudinal model that characterizes the progression of the atrophy of each region of the brain. This atrophy evolution can then be visualized thanks to a neurodegeneration simulation generated by the trained model by sampling missing intermediate values.

3.7 Benchmarks Conducted in the Literature

This section describes studies that compared several interpretability methods. We separated evaluations based on metrics from those which are purely qualitative. Indeed, even if the interpretability metrics are not mature yet, it is essential to try to measure quantitatively the difference between methods rather than to only rely on human perception, which may be biased.

3.7.1 Quantitative Evaluations

Eitel and Ritter [37] tested the robustness of four methods: standard perturbation, gradient \odot input, guided back-propagation, and LRP. To evaluate these methods, the authors trained ten times the same model with a random initialization and generated attribution maps for each of the ten runs. For each method, they exhibited significant differences between the averaged true positives/negatives attribution maps of the ten runs. To quantify this variance, they computed the ℓ_2 -norm between the attribution maps and determined for each model the brain regions with the highest attribution. They concluded that LRP and guided back-propagation were the most consistent methods, both in terms of distance between attribution maps and most relevant brain regions. However, this study makes a strong assumption: to draw these conclusions, the network should provide stable interpretations across retrainings. Unfortunately, Thibeau-Sutre et al. [51] showed that the study of the robustness of the interpretability method and of the network should be done separately, as their network retraining was not robust. Indeed, they first showed that the interpretability method they chose (optimized perturbation) was robust according to different criteria, and then they observed that network retraining led to different attribution maps. The robustness of an interpretability method thus cannot be assessed from the protocol described in [37]. Moreover, the fact that guided back-propagation

is one of the most stable method meets the results of [6], who observed that guided back-propagation always gave the same result independently from the weights learned by a network (*see* Subheading 4.1).

Böhle et al. [34] measured the benefit of LRP with β -rule compared to guided back-propagation by comparing the intensities of the mean attribution map of demented patients and the one of cognitively normal controls. They concluded that LRP allowed a stronger distinction between these two classes than guided back-propagation, as there was a greater difference between the mean maps for LRP. Moreover, they found a stronger correlation between the intensities of the LRP attribution map in the hippocampus and the hippocampal volume than for guided back-propagation. But as [6] demonstrated that guided back-propagation has serious flaws, it does not allow drawing strong conclusions.

Nigri et al. [45] compared the standard perturbation method to a swap test (*see* Subheading 3.4) using two properties: the continuity and the sensitivity. The continuity property is verified if two similar input images have similar explanations. The sensitivity property affirms that the most salient areas in an explanation map should have the greater impact in the prediction when removed. The authors carried out experiments with several types of models, and both properties were consistently verified for the swap test, while the standard perturbation method showed a significant absence of continuity and no conclusive fidelity values [45].

Finally, Rieke et al. [48] compared four visualization methods: standard back-propagation, guided back-propagation, standard perturbation, and brain area perturbation. They computed the Euclidean distance between the mean attribution maps of the same class for two different methods and observed that both gradient methods were close, whereas brain area perturbation was different from all others. They concluded that as interpretability methods lead to different attribution maps, one should compare the results of available methods and not trust only one attribution map.

3.7.2 Qualitative Evaluations

Some works compared interpretability methods using a purely qualitative evaluation.

First, Eitel et al. [38] generated attribution maps using the LRP and gradient \odot input methods and obtained very similar results. This could be expected as it was shown that there is a strong link between LRP and gradient \odot input (*see* Subheading 2.3.2).

Dyrba et al. [36] compared DeconvNet, guided back-propagation, deep Taylor decomposition, gradient \odot input, LRP (with various rules), and Grad-CAM. The different methods roughly exhibited the same highlighted regions but with a

significant variability in focus, scatter, and smoothness, especially for the Grad-CAM method. These conclusions were derived from a visual analysis. According to the authors, LRP and deep Taylor decomposition delivered the most promising results with a highest focus and less scatter [36].

Tang et al. [49] compared two interpretability methods that seemed to have different properties: guided Grad-CAM would provide a fine-grained view of feature salience, whereas standard perturbation highlights the interplay of features among classes. A similar conclusion was drawn by Rieke et al. [48].

3.7.3 Conclusions from the Benchmarks

The most extensively compared method is LRP, and each time it has been shown to be the best method compared to others. However, its equivalence with gradient \odot input for networks using ReLU activations still questions the usefulness of the method, as gradient \odot input is much easier to implement. Moreover, the studies reaching this conclusion are not very insightful: [37] may suffer from methodological biases; [34] compared LRP only to guided back-propagation, which was shown to be irrelevant [6]; and [36] only performed a qualitative assessment.

As proposed in conclusion by Rieke et al. [48], a good way to assess the quality of interpretability methods could be to produce some form of ground truth for the attribution maps, for example, by implementing simulation models that control for the level of separability or location of differences.

4 Limitations and Recommendations

Many methods have been proposed for interpretation of deep learning models. The field is not mature yet, and none of them has become a standard. Moreover, a large panel of studies has been applied to neuroimaging data, but the value of the results obtained from the interpretability methods is often still not clear. Furthermore, many applications suffer from methodological issues, making their results (partly) irrelevant. In spite of this, we believe that using interpretability methods is highly useful, in particular to spot cases where the model exploits biases in the data set.

4.1 Limitations of the Methods

It is not often clear whether the interpretability methods really highlight features relevant to the algorithm they interpret. This way, Adebayo et al. [6] showed that the attribution maps produced by some interpretability methods (guided back-propagation and guided Grad-CAM) may not be correlated at all with the weights learned by the network during its training procedure. They prove it with a simple test called “cascading randomization.” In this test, the weights of a network trained on natural images are randomized layer per layer, until the network is fully randomized. At each step,

they produce an attribution map with a set of interpretability methods to compare it to the original ones (attribution maps produced without randomization). In the case of guided back-propagation and guided Grad-CAM, all attribution maps were identical, which means that the results of these methods were independent of the training procedure.

Unfortunately, this type of failures does not only affect interpretability methods but also the metrics designed to evaluate their reliability, which makes the problem even more complex. Tomsett et al. [52] investigated this issue by evaluating interpretability metrics with three properties:

- **Inter-rater interpretability** assesses whether a metric always rank different interpretability methods in the same way for different samples in the data set.
- **Inter-method reliability** checks that the scores given by a metric on each saliency method fluctuate in the same way between images.
- **Internal consistency** evaluates if different metrics measuring the same property (e.g., fidelity) produce correlated scores on a set of attribution maps.

They concluded that the investigated metrics were not reliable, though it is difficult to know the origin of this unreliability due to the tight coupling of model, interpretability method, and metric.

4.2 Methodological Advice

Using interpretability methods is more and more common in medical research. Even though this field is not yet mature and the methods have limitations, we believe that using an interpretability method is usually a good thing because it may spot cases where the model took decisions from irrelevant features. However, there are methodological pitfalls to avoid and good practices to adopt to make a fair and sound analysis of your results.

You should first clearly state in your paper which interpretability method you use as there exist several variants for most of the methods (*see* Subheading 2), and its parameters should be clearly specified. Implementation details may also be important: for the Grad-CAM method, attribution maps can be computed at various levels in the network; for a perturbation method, the size and the nature of the perturbation greatly influence the result. The data on which methods are applied should also be made explicit: for a classification task, results may be completely different if samples are true positives or true negatives, or if they are taken from the train or test sets.

Taking a step back from the interpretability method and especially attribution maps is fundamental as they present several limitations [34]. First, there is no ground truth for such maps, which are usually visually assessed by authors. Comparing obtained results

with the machine learning literature is a good first step, but be aware that you will most of the time find a paper to support your findings, so we suggest to look at established clinical references. Second, attribution maps are usually sensitive to the interpretability method, its parameters (e.g., β for LRP), but also to the final scale used to display maps. A slight change in one of these variables may significantly impact the interpretation. Third, an attribution map is a way to measure the impact of pixels on the prediction of a given model, but it does not provide underlying reasons (e.g., pathological shape) or explain potential interactions between pixels. A given pixel might have a low attribution when considered on its own but have a huge impact on the prediction when combined with another. Fourth, the quality of a map strongly depends on the performance of the associated model. Indeed, low-performance models are more likely to use wrong features. However, even in this case, attribution maps may be leveraged, e.g., to determine if the model effectively relies on irrelevant features (such as visual artefacts) or if there are biases in the data set [53].

One must also be very careful when trying to establish new medical findings using model interpretations, as we do not always know how the interpretability methods react when applied to correlated features. Then even if a feature seems to have no interest for a model, this does not mean that it is not useful in the study of the disease (e.g., a model may not use information from the frontal lobe when diagnosing Alzheimer's disease dementia, but this does not mean that this region is not affected by the disease).

Finally, we suggest implementing different interpretability methods to obtain complementary insights from attribution maps. For instance, using LRP in addition to the standard back-propagation method provides a different type of information, as standard back-propagation gives the sensibility of the output with respect to the input, while LRP shows the contribution of each input feature to the output. Moreover, using several metrics allows a quantitative comparison between them using interpretability metrics (*see* Subheading 2.7).

4.3 Which Method Should I Choose?

We conclude this section on how to choose an interpretability method. Some benchmarks were conducted to assess the properties of some interpretability methods compared to others (*see* Subheading 3.7). Though these are good initiatives, there are still not enough studies (and some of them suffer from methodological flaws) to draw solid conclusions. This is why we give in this section some practical advice to the reader to choose an interpretability method based on more general concepts.

Before implementing an interpretability method, we suggest reviewing the following points to help you choose carefully.

- **Implementation complexity** Some methods are more difficult to implement than others and may require substantial coding efforts. However, many of them have already been implemented in libraries or GitHub repositories (e.g., [54]), so we suggest looking online before trying to re-implement them. This is especially true for model-agnostic methods, such as LIME, SHAP, or perturbations, for which no modification of your model is required. For model-specific methods, such as back-propagation ones, the implementation will depend on the model, but if its structure is a common one (e.g., regular CNN with feature extraction followed by a classifier), it is also very likely that an adequate implementation is already available (e.g., Grad-CAM on CNN in [54]).
- **Time cost** Computation time greatly differs from one method to another, especially when input data is heavy. For instance, perturbing high dimension images is time expensive, and it would be much faster to use standard back-propagation.
- **Method parameters** The number of parameters to set varies between methods, and their choice may greatly influence the result. For instance, the patch size, the step size (distance between two patches), as well as the type of perturbation (e.g., white patches or blurry patches) must be chosen for the standard perturbation method, while the standard back-propagation does not need any parameter. Thus, without prior knowledge on the interpretability results, methods with no or only a few parameters are a good option.
- **Literature** Finally, our last piece of advice is to look into the literature to determine the methods that have commonly been used in your domain of study. A highly used method does not guarantee its quality (e.g., guided back-propagation [6]), but it is usually a good first try.

To sum up, we suggest that you choose (or at least begin with) an interpretability method that is easy to implement, time efficient, with no parameters (or only a few) to tune, and commonly used. In the context of brain image analysis, we suggest using the standard back-propagation or Grad-CAM methods. Before using a method you do not know well, you should check that other studies did not show that this method is not relevant (which is the case for guided back-propagation or guided Grad-CAM) or that it is not equivalent to another method (e.g., LRP on networks with ReLU activation layers and \odot input).

Regarding interpretability metrics, there is no consensus in the community as the field is not mature yet. General advice would be to use different metrics and confront them to human observers, taking, for example, the methodology described in [1].

5 Conclusion

Interpretability of machine learning models is an important topic, in particular in the medical field. First, this is a natural need expressed by clinicians who are potential users of medical decision support systems. Moreover, it has been shown in many occasions that models with high performance can actually be using irrelevant features. This is dangerous because it means that they are exploiting biases in the training data sets and thus may dramatically fail when applied to new data sets or deployed in clinical routine.

Interpretability is a very active field of research and many approaches have been proposed. They have been extensively applied in neuroimaging and very often allowed highlighting clinically relevant regions of the brain that were used by the model. However, comparative benchmarks are not entirely conclusive, and it is currently not clear which approach is the most adapted for a given aim. In other words, it is very important to keep in mind that the field of interpretability is not yet mature. It is not yet clear which are the best methods or even if the most widely used approaches will still be considered a standard in the near future.

That being said, we still strongly recommend that a classification or regression model be studied with at least one interpretability method. Indeed, evaluating the performance of the model is not sufficient in itself, and the additional use of an interpretation method may allow detecting biases and models that perform well but for bad reasons and thus would not generalize to other settings.

Acknowledgements

The research leading to these results has received funding from the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute) and reference ANR-10-IAIHU-06 (Agence Nationale de la Recherche-10-IA Institut Hospitalo-Universitaire-6).

Appendices

A Short Reminder on Network Training Procedure

During the training phase, a neural network updates its weights to make a series of inputs match with their corresponding target labels:

1. *Forward pass* The network processes the input image to compute the output value.

2. *Loss computation* The difference between the true labels and the output values is computed according to a criterion (cross-entropy, mean squared error. . .). This difference is called the loss and should be as low as possible.
3. *Backward pass* For each learnable parameter of the network, the gradients with respect to the loss are computed.
4. *Weight update* Weights are updated according to the gradients and an optimizer rule (stochastic gradient descent, Adam, Adadelta. . .).

As a network is a composition of functions, the gradients of the weights of a layer l with respect to the loss can be easily obtained according to the values of the gradients in the following layers. This way of computing gradients layer per layer is called back-propagation.

***B Description of the
Main Brain Disorders
Mentioned in the
Reviewed Studies***

This appendix aims at shortly presenting the diseases considered by the studies reviewed in Subheading 3.

The majority of the studies focused on the classification of Alzheimer's disease (AD), a neurodegenerative disease of the elderly. Its pathological hallmarks are senile plaques formed by amyloid- β protein and neurofibrillary tangles that are tau protein aggregates. Both can be measured in vivo using either PET imaging or CSF biomarkers. Several other biomarkers of the disease exist. In particular, atrophy of gray and white matter measured from T1w MRI is often used, even though it is not specific to AD. There is strong and early atrophy in the hippocampi that can be linked to the memory loss, even though other clinical signs are found and other brain areas are altered. The following diagnosis statuses are often used:

- **AD** refers to demented patients.
- **CN** refers to cognitively normal participants.
- **MCI** refers to patients in with mild cognitive impairment (they have an objective cognitive decline, but it is not sufficient yet to cause a loss of autonomy).
- **Stable MCI** refers to MCI patients who stayed stable during a defined period (often three years).
- **Progressive MCI** refers to MCI patients who progressed to Alzheimer's disease during a defined period (often three years).

Most of the studies analyzed T1w MRI data, except [49] where the patterns of amyloid- β in the brain are studied.

Frontotemporal dementia is another neurodegenerative disease in which the neuronal loss dominates in the frontal and temporal lobes. Behavior and language are the most affected cognitive functions.

Parkinson's disease is also a neurodegenerative disease. It primarily affects dopaminergic neurons in the substantia nigra. A commonly used neuroimaging technique to detect this loss of dopaminergic neurons is the SPECT, as it uses a ligand that binds to dopamine transporters. Patients are affected by different symptoms linked to motor faculties such as tremor, slowed movements, and gait disorder but also sleep disorder, depression, and other symptoms.

Multiple sclerosis is a demyelinating disease with a neurodegenerative component affecting younger people (it begins between the ages of 20 and 50). It causes demyelination of the white matter in the brain (brain stem, basal ganglia, tracts near the ventricles), optic nerve, and spinal cord. This demyelination results in autonomic, visual, motor, and sensory problems.

Intracranial hemorrhage may result from a physical trauma or non-traumatic causes such as a ruptured aneurysm. Different subtypes exist depending on the location of the hemorrhage.

Autism is a spectrum of neurodevelopmental disorders affecting social interaction and communication. Diagnosis is done based on clinical signs (behavior), and the patterns that may exist in the brain are not yet reliably described as they overlap with the neurotypical population.

Some brain characteristics that may be related to brain disorders and detected in CT scans were considered in the data set CQ500:

- **Midline Shift** is a shift of the center of the brain past the center of the skull.
- **Mass Effect** is caused by the presence of an intracranial lesion (e.g., a tumor) that is compressing nearby tissues.
- **Calvarial Fractures** are fractures of the skull.

Finally, one study [33] learned to predict the age of cognitively normal patients. Such algorithm can help in diagnosing brain disorders as patients will have a greater brain age than their chronological age, and then it establishes that a participant is not in the normal distribution.

References

1. Ribeiro MT, Singh S, Guestrin C (2016) 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 – KDD '16, ACM Press, San Francisco, pp 1135–1144. <https://doi.org/10.1145/2939672.2939778>
2. Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: 2017 IEEE international conference on computer vision (ICCV), pp 3449–3457. <https://doi.org/10.1109/ICCV.2017.371>
3. DeGrave AJ, Janizek JD, Lee SI (2021) AI for radiographic COVID-19 detection selects

- shortcuts over signal. *Nat Mach Intell* 3(7): 610–619. <https://doi.org/10.1038/s42256-021-00338-7>
4. Lipton ZC (2018) The myths of model interpretability. *Commun ACM* 61(10):36–43. <https://doi.org/10.1145/3233231>
 5. Xie N, Ras G, van Gerven M, Doran D (2020) Explainable deep learning: a field guide for the uninitiated. *arXiv:2004.14545* [cs, stat] 2004.14545
 6. Adebayo J, Gilmer J, Muelly M, Goodfellow I, Hardt M, Kim B (2018) Sanity checks for saliency maps. In: *Advances in Neural Information Processing Systems*, pp 9505–9515
 7. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) *Advances in Neural information processing systems*, vol 25. Curran Associates, pp 1097–1105
 8. Voss C, Cammarata N, Goh G, Petrov M, Schubert L, Egan B, Lim SK, Olah C (2021) Visualizing weights. *Distill* 6(2):e00024.007. <https://doi.org/10.23915/distill.00024.007>
 9. Olah C, Mordvintsev A, Schubert L (2017) Feature visualization. *Distill* 2(11):e7. <https://doi.org/10.23915/distill.00007>
 10. Simonyan K, Vedaldi A, Zisserman A (2013) Deep inside convolutional networks: visualising image classification models and saliency maps. *arXiv:1312.6034* [cs] 1312.6034
 11. Shrikumar A, Greenside P, Shcherbina A, Kundaje A (2017) Not just a black box: learning important features through propagating activation differences. *arXiv:1605.01713* [cs] 1605.01713
 12. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for Simplicity: the all convolutional net. *arXiv:1412.6806* [cs] 1412.6806
 13. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2015) Learning deep features for discriminative localization. *arXiv:1512.04150* [cs] 1512.04150
 14. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-CAM: visual explanations from deep networks via gradient-based localization. In: *2017 IEEE international conference on computer vision (ICCV)*, pp 618–626. <https://doi.org/10.1109/ICCV.2017.74>
 15. Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS One* 10(7):e0130140. <https://doi.org/10.1371/journal.pone.0130140>
 16. Samek W, Binder A, Montavon G, Lapuschkin S, Müller KR (2017) Evaluating the visualization of what a deep neural network has learned. *IEEE Trans Neural Netw Learn Syst* 28(11):2660–2673. <https://doi.org/10.1109/TNNLS.2016.2599820>
 17. Montavon G, Lapuschkin S, Binder A, Samek W, Müller KR (2017) Explaining non-linear classification decisions with deep Taylor decomposition. *Pattern Recogn* 65:211–222. <https://doi.org/10.1016/j.patcog.2016.11.008>
 18. Montavon G, Samek W, Müller KR (2018) Methods for interpreting and understanding deep neural networks. *Digit Signal Process* 73:1–15. <https://doi.org/10.1016/j.dsp.2017.10.011>
 19. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) *Computer vision – ECCV 2014. Lecture notes in computer science*. Springer, Berlin, pp 818–833
 20. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: *Proceedings of the 31st international conference on neural information processing systems, NIPS’17*. Curran Associates, Red Hook, pp 4768–4777
 21. Frosst N, Hinton G (2017) Distilling a Neural network into a soft decision tree. *arXiv:1711.09784* [cs, stat] 1711.09784
 22. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhutdinov R, Zemel R, Bengio Y (2016) Show, attend and tell: neural image caption generation with visual attention. *arXiv:1502.03044* [cs] 1502.03044
 23. Wang F, Jiang M, Qian C, Yang S, Li C, Zhang H, Wang X, Tang X (2017) Residual attention network for image classification. In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, Honolulu, pp 6450–6458. <https://doi.org/10.1109/CVPR.2017.683>
 24. Ba J, Mnih V, Kavukcuoglu K (2015) Multiple object recognition with visual attention. *arXiv:1412.7755* [cs] 1412.7755
 25. Yeh CK, Hsieh CY, Suggala A, Inouye DI, Ravikumar PK (2019) On the (In) fidelity and sensitivity of explanations. In: Wallach H, Larochelle H, Beygelzimer A, d’Alché-Buc F, Fox E, Garnett R (eds) *Advances in neural information processing systems*, vol 32. Curran Associates, pp 10967–10978

26. Cecotti H, Gräser A (2011) Convolutional neural networks for P300 detection with application to brain-computer interfaces. *IEEE Trans on Pattern Anal Mach Intell* 33(3): 433–445. <https://doi.org/10.1109/TPAMI.2010.125>
27. Oh K, Chung YC, Kim KW, Kim WS, Oh IS (2019) Classification and visualization of Alzheimer's disease using volumetric convolutional neural network and transfer learning. *Sci Rep* 9(1):1–16. <https://doi.org/10.1038/s41598-019-54548-6>
28. Abrol A, Bhattarai M, Fedorov A, Du Y, Plis S, Calhoun V (2020) Deep residual learning for neuroimaging: an application to predict progression to Alzheimer's disease. *J Neurosci Methods* 339:108701. <https://doi.org/10.1016/j.jneumeth.2020.108701>
29. Biffi C, Cerrolaza J, Tarroni G, Bai W, De Marvao A, Oktay O, Ledig C, Le Folgoc L, Kamnitsas K, Doumou G, Duan J, Prasad S, Cook S, O'Regan D, Rueckert D (2020) Explainable anatomical shape analysis through deep Hierarchical generative models. *IEEE Trans Med Imaging* 39(6):2088–2099. <https://doi.org/10.1109/TMI.2020.2964499>
30. Martinez-Murcia FJ, Ortiz A, Gorriz JM, Ramirez J, Castillo-Barnes D (2020) Studying the manifold structure of Alzheimer's disease: a deep learning approach using convolutional autoencoders. *IEEE J Biomed Health Inf* 24(1):17–26. <https://doi.org/10.1109/JBHI.2019.2914970>
31. Leming M, Górriz JM, Suckling J (2020) Ensemble deep learning on large, mixed-site fMRI datasets in autism and other tasks. *Int J Neural Syst* 2050012. <https://doi.org/10.1142/S0129065720500124>, 2002.07874
32. Bae J, Stocks J, Heywood A, Jung Y, Jenkins L, Katsaggelos A, Popuri K, Beg MF, Wang L (2019) Transfer learning for predicting conversion from mild cognitive impairment to dementia of Alzheimer's type based on 3D-convolutional neural network. *bioRxiv*. <https://doi.org/10.1101/2019.12.20.884932>
33. Ball G, Kelly CE, Beare R, Seal ML (2021) Individual variation underlying brain age estimates in typical development. *Neuroimage* 235:118036. <https://doi.org/10.1016/j.neuroimage.2021.118036>
34. Böhle M, Eitel F, Weygandt M, Ritter K, on botADNI (2019) Layer-wise relevance propagation for explaining deep neural network decisions in MRI-based Alzheimer's disease classification. *Front Aging Neurosci* 10(JUL). <https://doi.org/10.3389/fnagi.2019.00194>
35. Burduja M, Ionescu RT, Verga N (2020) Accurate and efficient intracranial hemorrhage detection and subtype classification in 3D CT scans with convolutional and long short-term memory neural networks. *Sensors* 20(19): 5611. <https://doi.org/10.3390/s20195611>
36. Dyrba M, Pallath AH, Marzban EN (2020) Comparison of CNN visualization methods to aid model interpretability for detecting Alzheimer's disease. In: Tolxdorff T, Deserno TM, Handels H, Maier A, Maier-Hein KH, Palm C (eds) *Bildverarbeitung für die Medizin 2020*, Springer Fachmedien, Wiesbaden, Informatik aktuell, pp 307–312. https://doi.org/10.1007/978-3-658-29267-6_68
37. Eitel F, Ritter K (2019) Testing the robustness of attribution methods for convolutional neural networks in MRI-based Alzheimer's disease classification. In: *Interpretability of machine intelligence in medical image computing and multimodal learning for clinical decision support*. Lecture notes in computer science. Springer, Cham, pp 3–11. https://doi.org/10.1007/978-3-030-33850-3_1
38. Eitel F, Soehler E, Bellmann-Strobl J, Brandt AU, Ruprecht K, Giess RM, Kuchling J, Assever S, Weygandt M, Haynes JD, Scheel M, Paul F, Ritter K (2019) Uncovering convolutional neural network decisions for diagnosing multiple sclerosis on conventional MRI using layer-wise relevance propagation. *Neuroimage: Clinical* 24:102003. <https://doi.org/10.1016/j.nicl.2019.102003>
39. Fu G, Li J, Wang R, Ma Y, Chen Y (2021) Attention-based full slice brain CT image diagnosis with explanations. *Neurocomputing* 452: 263–274. <https://doi.org/10.1016/j.neucom.2021.04.044>
40. Gutiérrez-Becker B, Wachinger C (2018) Deep multi-structural shape analysis: application to neuroanatomy. In: *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, LNCS, vol 11072, pp 523–531. https://doi.org/10.1007/978-3-030-00931-1_60
41. Hu J, Qing Z, Liu R, Zhang X, Lv P, Wang M, Wang Y, He K, Gao Y, Zhang B (2021) Deep learning-based classification and voxel-based visualization of frontotemporal dementia and Alzheimer's disease. *Front Neurosci* 14. <https://doi.org/10.3389/fnins.2020.626154>
42. Jin D, Zhou B, Han Y, Ren J, Han T, Liu B, Lu J, Song C, Wang P, Wang D, Xu J, Yang Z, Yao H, Yu C, Zhao K, Wintermark M, Zuo N,

- Zhang X, Zhou Y, Zhang X, Jiang T, Wang Q, Liu Y (2020) Generalizable, reproducible, and neuroscientifically interpretable imaging biomarkers for Alzheimer's disease. *Adv Sci* 7(14):2000675. <https://doi.org/10.1002/adv.202000675>
43. Lee E, Choi JS, Kim M, Suk HI (2019) Alzheimer's disease neuroimaging initiative toward an interpretable Alzheimer's disease diagnostic model with regional abnormality representation via deep learning. *NeuroImage* 202: 116113. <https://doi.org/10.1016/j.neuroimage.2019.116113>
 44. Magesh PR, Myloth RD, Tom RJ (2020) An explainable machine learning model for early detection of Parkinson's disease using LIME on DaTSCAN imagery. *Comput Biol Med* 126:104041. <https://doi.org/10.1016/j.compbimed.2020.104041>
 45. Nigri E, Ziviani N, Cappabianco F, Antunes A, Veloso A (2020) Explainable deep CNNs for MRI-based diagnosis of Alzheimer's disease. In: Proceedings of the international joint conference on neural networks. <https://doi.org/10.1109/IJCNN48605.2020.9206837>
 46. Qiu S, Joshi PS, Miller MI, Xue C, Zhou X, Karjadi C, Chang GH, Joshi AS, Dwyer B, Zhu S, Kaku M, Zhou Y, Alderazi YJ, Swaminathan A, Kedar S, Saint-Hilaire MH, Auerbach SH, Yuan J, Sartor EA, Au R, Kolachalama VB (2020) Development and validation of an interpretable deep learning framework for Alzheimer's disease classification. *Brain: J Neurol* 143(6):1920–1933. <https://doi.org/10.1093/brain/awaa137>
 47. Ravi D, Blumberg SB, Ingala S, Barkhof F, Alexander DC, Oxtoby NP (2022) Degenerative adversarial neuroimage nets for brain scan simulations: application in ageing and dementia. *Med Image Anal* 75:102257. <https://doi.org/10.1016/j.media.2021.102257>
 48. Rieke J, Eitel F, Weygandt M, Haynes JD, Ritter K (2018) Visualizing convolutional networks for MRI-based diagnosis of Alzheimer's disease. In: Understanding and interpreting machine learning in medical image computing applications. Lecture notes in computer science. Springer, Cham, pp 24–31. https://doi.org/10.1007/978-3-030-02628-8_3
 49. Tang Z, Chuang KV, DeCarli C, Jin LW, Beckett L, Keiser MJ, Dugger BN (2019) Interpretable classification of Alzheimer's disease pathologies with a convolutional neural network pipeline. *Nat Commun* 10(1):1–14. <https://doi.org/10.1038/s41467-019-10212-1>
 50. Wood D, Cole J, Booth T (2019) NEURODRAM: a 3D recurrent visual attention model for interpretable neuroimaging classification. arXiv:191004721 [cs, stat] 1910.04721
 51. Thibeau-Sutre E, Colliot O, Dormont D, Burgos N (2020) Visualization approach to assess the robustness of neural networks for medical image classification. In: Medical imaging 2020: image processing, international society for optics and photonics, vol 11313, p 113131J. <https://doi.org/10.1117/12.2548952>
 52. Tomsett R, Harborne D, Chakraborty S, Gurrum P, Preece A (2020) Sanity checks for saliency metrics. *Proc AAAI Conf Artif Intell* 34(04):6021–6029. <https://doi.org/10.1609/aaai.v34i04.6064>
 53. Lapuschkin S, Binder A, Montavon G, Muller KR, Samek W (2016) Analyzing classifiers: fisher vectors and deep neural networks. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Las Vegas, pp 2912–2920. <https://doi.org/10.1109/CVPR.2016.318>
 54. Ozubulak U (2019) Pytorch cnn visualizations. <https://github.com/utkuozbulak/pytorch-cnn-visualizations>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution, and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third-party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

