Determining the Relevance of Features for Deep Neural Networks

Christian Reimers[1,2][0000−0003−1127−136X], Jakob Runge[2][0000−0002−0629−1772], and Joachim Denzler[1][0000−0002−3193−3300]

[1] Computer Vision Group, Friedrich Schiller University Jena, 07743 Jena, Germany
{christian.reimers,joachim.denzler}@uni-jena.de
[2] Institute of Data Science, German Aerospace Center, 07745 Jena, Germany
jakob.runge@dlr.de

Abstract. Deep neural networks are tremendously successful in many applications, but end-to-end trained networks often result in hard to understand black-box classifiers or predictors. In this work, we present a novel method to identify whether a specific feature is relevant to a classifier's decision or not. This relevance is determined at the level of the learned mapping, instead of for a single example. The approach does neither need retraining of the network nor information on intermediate results or gradients. The key idea of our approach builds upon concepts from causal inference. We interpret machine learning in a structural causal model and use Reichenbach's common cause principle to infer whether a feature is relevant. We demonstrate empirically that the method is able to successfully evaluate the relevance of given features on three real-life data sets, namely MS COCO, CUB200 and HAM10000.

Keywords: Explainable-AI, Structural Causal Model, Deep learning, Causality

1 Introduction

Deep neural networks have pushed the state-of-the-art in many tasks in computer vision, for example, image classification [15], semantic image segmentation [18] and object detection [24]. Because of their success in these tasks, deep neural networks are used in many applications such as transport [11], medicine [23], legal decisions [2] and earth system science [26].

To reach these impressive results, deep neural network architectures have become more and more complex. The complex deep architectures allow the networks to perform automatic feature selection [27]. While the automatic feature selection led to better performance, the resulting neural networks are opaque, and it is difficult to understand how the network reaches its decisions and how it will decide on new data. One open question is, for example, whether the network uses a specific feature to reach its decision.

In many tasks, it is important to understand which features are selected by a deep neural network. It is important not just in safety- and security-critical

tasks like autonomous driving or medicine, but also in responsible tasks like legal decisions. For a neural network to aid humans to come to informed decisions, it is often not enough to provide predictions, but the reasoning behind the decision is equally important. For example, in responsible decisions, like legal decisions, we need to make sure that neural networks are unbiased and do not discriminate against genders or minorities.

A simple and often used idea to decide if a feature is relevant to a response variable is correlation [14]. To determine the relevance of a feature to the prediction, we could use the output of the neural network as the response variable and calculate the correlation. Correlation is, however, for multiple reasons not well suited for deep neural networks. The first reason is that neural networks realize complicated, non-linear functions. In contrast, correlation measures only linear relationships. The second reason is that confounding of variables might lead to correlation between features and the prediction of the deep neural network, even if changes in the feature would not influence the prediction of the deep neural network and, hence, we should consider the feature to be irrelevant. An example, that demonstrates the problem of confounding can be found in Section 3.

To mitigate the problem of confounding, many researchers have used the gradient between the output and the input of a neural network instead of the correlation [36, 29, 16, 21]. This solves the problem of confounding but leads to new challenges. First, these methods approximate the gradient of the deep neural network on the level of single examples. While this is very useful in many situations, we are interested in explanations on the level of the learned mapping. On this level, the abovementioned methods do not provide insight. For example, in a classification system, the above methods can highlight which part of an image is relevant for the classification of this image, while our method can answer the question if some feature is, in general, relevant to the decision of the classifier. Furthermore, if we want to understand the relevance of an image region or an intermediate result of the neural network, we can use gradient information. However, we might be interested in features that are themselves function of the input. If we cannot identify a neuron that calculates this feature, we cannot calculate a gradient. Consequently, if we are interested in a feature that is a function of the input, we cannot use methods that depend on gradients.

Additionally, to calculate the gradients, we need access to the inner structure and the weights of the neural network. In the case where we have access to these values, we speak of a white-box model. Contrarily, if we do not have any access to these values, we speak of a black-box model.

In this paper, we present a novel method to determine whether a feature is relevant to a deep neural network. The key advantages of this novel method are:

- Our method is applicable to features that are functions of the input.
- It can handle non-linearity by using statistical dependence instead of correlation.
- It can handle confounding.
- Our method can be applied to black-box models, neither any retraining nor any information on intermediate results or gradients is necessary.

To the best of our knowledge our method is the only one that combines all of these advantages.

We achieve this by capturing the fundamental structure of deep learning into a structural causal model [22] (SCM). In this SCM, we calculate the statistical dependence between the feature and the prediction of the deep neural network. We use Reichenbach's common cause principle [25] to determine whether a feature is relevant to the decision of a deep neural network. We demonstrate that the method can be used to validate classifiers in real-live situations. In the experiments on MS COCO we compare two classifiers on which features are relevant to their decisions. We find that the classifier from [4] is less dependent on the position of the objects than the classifier from [5]. In an experiment on HAM10000 we show that the symmetry of the shape is used by a classifier to identify seborrheic keratosis but not to identify melanoma.

2 Related Work

The class of method that is used most often to understand deep neural networks is saliency maps. Saliency maps are functions that map every pixel of the input onto a relevance score. This relevance score highlights areas of the input. It is calculated in one of three ways.

The first way is to use the gradient of the output of the neural network depending on the input. Small changes in pixels with a large gradient lead to large changes in the output of the neural network. Hence, these pixels are considered to be more relevant. Examples can be found in [29] or [28].

The second way is to use the value of pixels and the value of activations in intermediate layers in addition to the gradient information. The relevance is propagated through the different layers similar to a Taylor-approximation. For examples see [16], [20] or [21].

Both of these ways use intermediate activations or gradient information and can, therefore, only be applied in the white-box case. Our method does not require any information on intermediate results or gradients. Hence, our method can be applied to black-box models and is not even restricted to neural networks.

Further, the first derivative is not an obvious measure for relevance in non-linear function. Only little theoretical justification exists and multiple papers advice for caution when interpreting saliency maps. For example, [1] reports that randomly-initialized neural networks produce saliency maps that are visually and quantitatively similar to those produced by deep neural networks with learned weights. Further, [13] reports that saliency maps fail to attribute correctly when a constant vector shift is applied. In contrast, we use the framework of causality and Reichenbach's common cause principle to provide a rich theoretical explanation for the method proposed in this paper.

The third way is based on intervention. We start by recording the prediction for an image. Then patches of the image are replaced and the change in the prediction is taken as the relevance of the patch. Patches can be replaced by

simple noise boxes [36], noisy versions of the original patch [6], by similar patches from other images [38], or by training a generative model [8].

Similar to gradients, an intervention can be performed on the input or on intermediate neurons. If we want to intervene on a feature that is a function of the input, and we cannot identify a neuron that calculates this function, it might be unclear how to intervene on it. In Section 4.4, we investigate the symmetry of skin lesions as a feature. To alter the symmetry of a skin lesion in a picture, one would need to replace significant parts of the image, which would naturally change the area, the mean color, the variance in color, the length of the border, the shape of the border and many more features. Further, there are infinitely many ways to change, for example, a symmetric skin lesion to a non-symmetric one. Since these ways might influence the aforementioned features differently, the outcome of the investigation will be different as well. Our method can be used in this situation. In contrast, the above way of calculating a saliency map can only be used if the feature is a part of the input or an intermediate neuron of the neural network.

Another key difference between all saliency maps and our method is that saliency maps explain decisions for individual inputs. In contrast, our method generates an explanation of the level of the learned mapping. Hence, we think that our method is a valuable addition and can be used together with saliency maps to get a more complete understanding of classifiers.

The authors of [12] present TCAV, a method to understand a neural network on the level of the learned mapping. They use images that represent a certain feature. An SVM is trained on intermediate representations extracted from the neural network. To determine whether the feature is relevant to the deep neural network's decision, they calculate whether the gradient of the deep neural network is in the same direction as the gradient of the SVM. Similar to our method, TCAV generates explanations on the level of the learned mapping, in contrast to our method that can be applied to black-box models, TCAV needs intermediate results of the neural network. Additionally, [8] reports that TCAV cannot resolve confounding.

3 Method

In this section, we explain how we use notation from the field of causal inference and Reichenbach's common cause principle to determine whether a feature is relevant to a classifier's decision. We reduce deep learning to a basic level. We identify the main elements necessary and arrange them into an SCM and subsequently into a graphical model. For each step of this process, we give a formal explanation and an illustration using an example. We will start by giving an introduction to this example. Then, first, we reduce the fundamental idea of deep learning to a pair of a training function and an inference function. Second, we explain which additional variables are needed to build this minimal formulation of deep learning into an SCM and, third, which functions connect these variables.

Fig. 1. From left to right, an example for the first class, an example for the second class, and all 2×3 patterns that appear in the first class but not in the second class are displayed.

Fourth, we explain why Reichenbach's common cause principle is applicable in this situation and how we use it to determine if a feature is relevant.

To this end, we use the notation of structural causal models (SCM) from [22]. An SCM is a model in which every variable is given as a function of other variables and independent noise. It can be built into a directed graph by using the variables as vertices and creating an edge from a variable A to a variable B if the function determining the value of B takes the variable A as an input. In this case, the authors of [22] say that A *causes* B.

The following is used as a vivid example for all definitions given in this section. The task is to distinguish between two classes of images. The images are each composed of a box for class one or a cross for class two and additive noise. Images of the base classes can be seen in Figure 1. The dark pixels have a value of 0.0, and the light pixels have a value of 0.9, the additive noise is for each pixel independently distributed according to a uniform distribution on the interval $[0, 0.1]$. The task is to determine whether the base image is the box or the cross. We created a training set from 10000 images.

In Figure 1, we also listed all 3×2 patterns that can be found in images of the first class but not in images of the second class. Note that each of these patterns alone is enough to distinguish the two classes. Since all of these patterns appear in all images of the first class and not in any image of the second class, the class is a confounding factor for the existence of these patterns and the prediction of the classifier. No matter which one of these patterns a classifier actually detects, the correlation between the existence of all of these patterns and the output of the classifier is high. Therefore, it is hard to evaluate which of these features is used by the classifier.

As the classifier, we use a convolutional neural network. The network has one convolutional layer with one kernel of size 3×2, followed by a global max-pooling layer. Due to this architecture, the classifier is forced to select a single discriminative pattern of size at most 3×2. A further advantage is that we can check the only kernel to verify which pattern was selected by the neural network.

The classifier is trained using stochastic gradient descent for ten epochs. The accuracy on a testing set sampled from the same distribution was 1.0.

To build an SCM, we, first, reduce the basic idea of deep learning to a pair (T, F) of a training function and an inference function.

The first function T is the training function. We denote the inputs of a deep neural network by $I \in \mathcal{I}$ and the corresponding label by $Y_I \in \mathcal{Y}$. The training function takes a set of labeled inputs $\{(I, Y_I)\}$, the training set TS, as the input and maps it onto a set of weights W. Most training algorithms, like stochastic gradient descent, do not determine the weights of the deep neural networks deterministically. Therefore, we define for a probability space Ω

$$T : \mathcal{P}\left((\mathcal{I} \times \mathcal{Y})\right) \times \Omega \to \mathbb{R}^m \qquad \{(I, Y_I)\}, \omega \mapsto W. \tag{1}$$

Here $\mathcal{P}$ denotes the power set. In the example defined above, the function T is given by the stochastic gradient descent used for training. The training set TS in our example is the set of 10000 examples and the weights W are the seven weights of the convolutional layer, six for the kernel and one for the bias.

The second function F is the inference function. This function takes an unlabeled input I and the weights W calculated by the training function T as the input and maps them onto a prediction P

$$F : \mathcal{I} \times \mathbb{R}^m \to \mathbb{R}^d \qquad I, W \mapsto P. \tag{2}$$

Here d is the dimension of the output. For a classifier with d classes, the output of a deep neural network is a d-dimensional vector containing the scores for each class. In the example, the function F is given by the neural network. The prediction P is given by the output of the neural network. Note that the output of the neural network is not in $\mathcal{Y}$ but in $\mathbb{R}$. The network outputs a positive value for images of class "box" and negative values for images of class "cross."

Second, we identify all random variables that are part of the SCM. These variables are given by the training set TS, the weights W, and the prediction P, which are explained above. Additionally, we need three more variables, namely the feature of interest X, the set $\bar{X}$ of all features independent of X, and the ground truth GT. We continue with a brief explanation of these variables.

The random variable X can denote any feature of interest. It can be continuous as in Section 4.2 or discrete as in Section 4.4.It is not important whether we determine the value from the image or know it from any other source. In our example, the feature X denotes the maximum over the convolution of a pattern (light squares have value one, dark squares have value minus one) with the image. A high value can be understood as the pattern appears in the image, while a low value indicates that the pattern does not appear in the image. For this example, we decide to investigate the bottom right corner pattern that is displayed in the middle of the right graph in Figure 2.

Since one of the advantages of deep neural networks is automatic feature selection, we are sure that the neural network bases its prediction on some features extracted from the input. We denote the set of all possible features independent of X by $\bar{X}$. Later, we will see that we never need to determine $\bar{X}$ further than

this definition. We neither need the value of these features nor do we need to determine which features are part of $\bar{X}$. This restriction to orthogonal features has the following reason. Imagine a situation in which the classifier calculates $f(x^3)$. One could argue that it uses x^3 or that it uses x as the relevant feature. It is neither possible nor desirable to decline one of these options. No method can or should determine one of these as more relevant. Additionally, the method will identify features as relevant if they contain a relevant feature. This is desirable since $f(x^3)$ can be calculated from (x, y) and hence, (x, y) should be identified as relevant. However, since we want to understand which feature is the input to an unknown function, we can identify X only up to a class of features. If the function is $f(X)$, we cannot distinguish it from, for example, $f(\log(\exp(X)))$ and hence, we cannot distinguish the features X from $\exp(X)$. More specific, we cannot distinguish features that are invertible mappings of each other. In our example, the set $\bar{X}$ contains all other patterns. It also contains all other numerical features of the images, for example, its mean and its variance.

Finally, the ground truth GT describes the distribution from which X, $\bar{X}$ and TS, parameterized by the label. In our example 'box' and 'cross'.

Third, to identify the SCM, we need to find all functions that connect these variables. Besides the training function and the inference function defined above, we identify two sampling processes. On the one hand, the process that samples the training set from the ground truth distributions and provides the labeled test set, on the other hand, the process that samples single unlabeled examples from the ground truth distribution over all possible examples. Since all features are deterministically determined by the sampled example, we can understand the second sampling function as sampling the features from the ground truth instead of sampling an example. Both of these sampling processes exist in our example.

Fourth, we use the framework of causal inference to build the graphical model and use Reichenbach's common cause principle to determine whether the feature X is relevant to the prediction of the deep neural network.

The graphical model created from the SCM described in this section can be found in Figure 2. The sampling processes sample TS, or X and $\bar{X}$ from GT. The training function maps the training set TS on the weights W, the inference function F determines the prediction P from the weights W and some features from the set $\bar{X}$. The remaining open question is, whether the inference function takes X as an input or, in other words, whether X *causes* P.

The way we identify whether a variable is causing another variable is Reichenbach's common cause principle (RCCP) [25]. This principle states that there is no correlation without causation. Since we are handling non-linear relationships, we look at statistical dependence instead of correlation. More formally, RCCP states that if two variables A and B are dependent, than there exists a variable C that *causes* A and B. In particular, C can be identical to A or B meaning that A *causes* B or B *causes* A.

We use RCCP to determine whether the feature X *causes* the prediction P. First, we want to rule out that a third variable *causes* X and P. From the

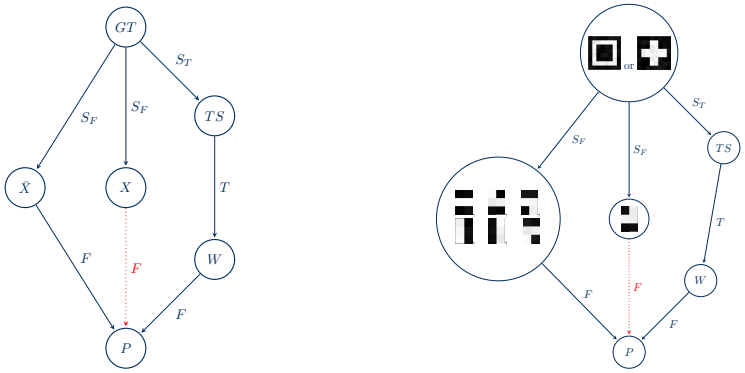

Fig. 2. On the left, the general graph of our method is displayed. On the right, we show the graph for the example.

graphical model in Figure 2, we see that the only variable that can *cause* X and P is GT. To eliminate this possibility, we condition on GT. Second, we can rule out the possibility that the prediction P cases X because the prediction P is only calculated after the feature X is sampled. Hence, if we find a statistical dependence between X and P after conditioning on GT

$$X \not\perp\!\!\!\perp P \,|GT, \tag{3}$$

we know that X *causes* P.

Note that we only need X, P, and GT for the calculations. In particular, we do not need $\bar{X}$. This allows us to handle all possible features, without ever calculating them.

To conclude, we say a feature X is relevant to the prediction P of a classifier or predictor if X is *causing* P in the sense of [22]. We use the word relevant instead of *causes* since it gives a better intuition in this exact application of the framework and to avoid confusion with the colloquial word "causes" or other definitions of causation such as the one presented by Granger in [9] or by Berkley in [3].

To illustrate further, we demonstrate the results for our example. We run the method seven times, once for each of the seven patterns illustrated in Figure 1. The results can be found in Table 1. Since we know that the relationships in this example are linear, we use partial correlation as the independence test. The significance level is set to 0.05. We find that all features are correlated to the output with a coefficient of determination greater than 0.975. Hence, the coefficient of determination is not suited to determine which feature is used. In contrast, our method can determine the correct pattern.

Table 1. Results for the running example. The investigated feature is the maximum of the convolution of the pattern and every image. The light spots have value one and the dark spots have value minus one. We report the coefficient of determination between the feature X and the prediction P. Second, we report the significance of a partial correlation test as a measure for $X \perp\!\!\!\perp P \,|GT$. The significance level is set to 0.05

Pattern	$r^2(X, P)$	$X \perp\!\!\!\perp P \,\|GT$	Equal to the kernel:
	1.000	**Yes** (p-value = 0.000)	**Yes**
	0.985	No (p-value = 0.608)	No
	0.985	No (p-value = 0.980)	No
	0.979	No (p-value = 0.949)	No
	0.979	No (p-value = 0.384)	No
	0.978	No (p-value = 0.422)	No
	0.978	No (p-value = 0.793)	No

4 Experiments

In this section, we describe four experiments that demonstrate that our proposed method works on real-live data sets and can identify whether features are used even if the feature is a function of the input rather than part of the input directly and neither a derivative nor an intervention can be calculated. To this end, we first check whether, for images from the MS COCO data set, the area and the position of an object in the image are relevant to recognizing the object in the image. Secondly, we check whether a state-of-the-art classifier on the CUB200 bird data set uses the same discriminative characteristics of a bird species an ornithologist would use. Thirdly, we investigate a system that checks whether a skin lesion is a melanoma or a seborrheic keratosis. The feature of interest for this example is the shape symmetry, a discrete score used by dermatologists to describe how symmetric the border of a skin lesion is.

4.1 Evaluating a Classifier on MS COCO

MS COCO [17] is a multi-label dataset of real-live images. We investigate the Multi-Label Graph Convolutional Networks (ML-GCN) presented by Chen et al. in [4]. In the first experiment, we investigate whether the size of the object given by the area of the image covered by this object is relevant to the classifier.

One of the advantages of MS COCO is that the object is not always in the center but can appear anywhere in the image. This should lead to a classifier independent of the position in which the objects appear in the image. This is for the ML-GCN additionally aided by data augmentation, including random horizontal flipping of images. We, therefore, investigate if the position of the object is relevant to the prediction of the ML-GCN and if the vertical position is more relevant than the horizontal position.

For this experiment, we used the pre-trained version of the ML-GCN provided by the authors. One advantage of our method is that we can use it in combination

with complex pre-trained classifiers. Neither is any retraining required nor do we need access to any weights or intermediate results of the classifier. This makes it easy and fast to apply. We conducted our experiments on the validation set of the 2014 challenge data set. As an independence test, we use the Randomized conditional Correlation Test (RCOT) [31]. Since we are testing many classes, we use a significance level of 0.001.

We run our method for the area of the object in the image and the position of the object in the image as the feature of interest. The area is calculated as its fraction of the image. If multiple instances of the object exist in the image, we consider the sum of all of their areas. We determine the area from the ground truth image segmentation. For the position, we consider the center of mass of the object. If more than one instance of the object is present in the image, we calculate the combined center of mass. For a more fine-grained analysis, we derive five features from the position of the object: The direction of the center of mass from the center of the image, the vertical position, the horizontal position, whether the object is in the right or left half and whether the object is in the top or bottom half of the image.

We expect, on the one hand, the area of the object to be relevant to the recognition score of the ML-GCN. Big objects should be easier to detect than small objects. On the other hand, we expect the position to be irrelevant to the recognition of the object. The information, whether the object is in the left or right half, should not be relevant since the images are randomly horizontally flipped during training. Further, we want an object recognition system to be independent of the position of the object.

We found that, for most of the 80 classes, the area of the object is relevant for the detection score. However, the area is irrelevant to the recognition of seven classes, namely "sheep," "elephant," "bear," "giraffe," "kite," "toaster," and "hairdryer". Further investigation sheds light on why the area of the object is irrelevant to some of these classes. On average, an object appears in 1267 images of the dataset. Instances of "hairdryer" and "toaster" appear in less than 75 images. This might lead to the network relying on context to recognize objects of these two classes. Interestingly, four of the five remaining classes are animals. Maybe, the neural network identifies only parts of the animal, e.g., the eyes, but further investigation is needed to unravel this further.

The results on the positional features can be found in the first column of Table 2. We find that despite the horizontal flipping, for one class, namely "mouse," it is relevant in which half of the image the object appears. As expected, due to the horizontal flipping, the horizontal position is much less relevant than the vertical position. The vertical half is relevant to 25 classes compared to 1 class for the horizontal half and the vertical position is relevant to 51 classes compared to 10 classes for the horizontal position. To summarize, we find that the classifier does not recognize objects independent of their position.

Note that the position of an object is a feature that cannot be highlighted by a saliency map.

Table 2. The results for the experiments on the MS COCO data set for the different classifiers. For each feature and each classifier, we report, for the detection of how many out of the 80 classes, the feature was relevant. Tables with the p-value for every class can be found in the appendix

Feature	ML-GCN	SRN
Area	73/80	69/80
Half(Horizontal)	1/80	1/80
Position(Horizontal)	10/80	16/80
Half(Vertical)	25/80	32/80
Position(Vertical)	51/80	54/80
Angle	24/80	31/80

Fig. 3. From left to right: An example of a white-crowned sparrow, an example from a white-throated sparrow, the segmentation map for the white throat markings, the segmentation map for the yellow lores.

4.2 Evaluating a Classifier on CUB200

In a second experiment, we use the method proposed in this paper to investigate a classifier that discriminates birds from the CUB200 [35] data set. We use the Inception-V3 large scale fine-grained classifier (LSFGC) presented by Cui et al. [5] that is pre-trained on the iNaturalist 2017 data set and fine-tuned on CUB200. We change the LSFGC to only differentiate between two bird species, the White-crowned sparrow and the White-throated sparrow. The two main differences between these birds are two yellow lores and white throat markings [33]. Example images for both of these birds, taken from the CUB200 data set, can be found in Figure 2. We check whether the LSFGC considers the area of the yellow lores, the area of the white throat markings, the color of the yellow lores, and the color of the white throat markings. We construct a feature out of each of these. For the area, we consider the fraction of the image that is covert by the yellow lores for the first feature and by the white throat markings for the second feature. As the color feature, we take the mean hue over the respective areas in the HSV color space. Segmentation maps for both of these features can be found in Figure 3. We use all 120 images from the two classes in the CUB200 data set. To determine independence, we again use the RCOT. We assume independence at a p-value below 0.05. The results can be found in Table 3.

Table 3. Results for the independence test that determine whether the yellow lores or the white throat markings are relevant to the LSFGC to distinguish white-crowned sparrows from white-throated sparrows

Feature	Area	Color
Yellow lores	No (p-value = 0.406)	No (p-value = 0.446)
white throat markings	No (p-value = 0.404)	No (p-value = 0.330)

We find that neither the size of the yellow or white throat markings nor their color is relevant to the classifier. This is not a surprising result since adversarial examples have demonstrated that neural networks often use complex, difficult to interpret features instead of the features considered relevant by human experts. [32, 7]

Note that a saliency map only highlight the region in the image that contains, for example, the yellow lores, while our method can distinguish between the area and the color of this region as different features.

4.3 Comparing Classifiers on MS COCO

We can also use the method presented in this work to compare classifiers beyond their performance on a test set. We compare the ML-GCN discussed above and the Spatial Regularization with Image-level Supervision classifier (SRN) presented by Zhu et al. [37]. To compare them, we use the abovementioned features. The results can be found in Table 2. We find that the ML-GCN considers the area of the actual object relevant for 73 classes, which is more often than the SRN, which considers it for only 69 classes. This indicates that the ML-GCN might be better at identifying the object while the SRN might rely more on the context to identify classes. Further, we find that the ML-GCN considers the position of the image to be relevant for fewer classes than the SRN. Since an ideal object detector would be independent of the position of the object, this analysis suggests that the ML-GCN should be preferred over the SRN.

4.4 Evaluating a Classifier on HAM10000

This fourth experiment is on the HAM10000 [34] data set, a data set of 10000 images of skin lesions. We investigate the classifier that won the best paper award at the ISIC Skin Image Analysis Workshop at the MICCAI 2018 [23]. We call this classifier SLA throughout this paper. The feature we investigate is the shape symmetry of the skin lesion.

The feature describes how many orthogonal lines can be found in the image of the skin lesion such that the shape of the skin lesion is symmetric with respect to all of these lines.

This feature is recognized to be important by the ABCD rule [30]. This rule is used by dermatologists and is known to be useful in distinguishing different

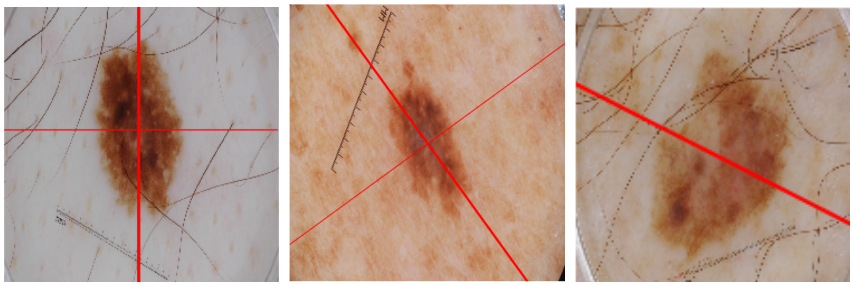

Fig. 4. Some example images for the shape symmetry feature. The red lines indicate the axis with respect to which the shape of the skin lesion is almost symmetric.

Table 4. We test whether the shape symmetry is relevant to the classification as melanoma or seborrheic keratosis, respectively. The significance level is set to 0.05

feature	melanoma	seborrheic keratosis
shape symmetry	No (p-value = 0.632)	**Yes** (p-value = 0.022)

classes of skin lesions. As discussed in Section 2, this feature can not be investigated using saliency maps or ablation methods.

We evaluate this feature on the ISIC 2017 Validation set. To calculate the shape symmetry, we use the implementation from [19]. The classifier outputs a score for two classes: "melanoma" and "seborrheic keratosis." For both of these classes, we determine whether the shape symmetry is relevant for the classification by the SLA. The result can be found in Table 4. We use RCOT for the independence tests. We consider the feature to be relevant if the p-value for the independence test is below 0.05.

We find that the shape symmetry is used by the classifier to determine whether a skin lesion is a seborrheic keratosis but not to determine whether a skin lesion is a melanoma.

5 Summary and Discussion

In this work, we presented a novel method to determine whether a feature is relevant to the prediction of a deep neural network. This method is built on the framework of causal-inference and has several key differences to other methods. Firstly, it determines the global behavior of a deep neural network instead of the behavior in single examples, and it can be used to evaluate the relevance of features that are functions of the input rather than part of the input directly. Hence, it complements other methods, such as saliency maps.

Our method can be applied to a black-box classifier. Neither any retraining nor any insight into intermediate results or gradients is needed. Hence, it is easy and fast to apply the method, and it does not lead to any decrease in the

performance of the classifier we want to understand. Additionally, since the only requirement to the classifier we made is that it can be formulated as a pair of two functions (T, F) our method can be applied to a vast range of classifiers. While, in this paper, we focus on deep neural networks, our method can also be applied to classical methods like kernel-SVM.

Since deep neural networks are opaque, it is difficult to evaluate the correctness of explanations for deep neural networks empirically. To this end, we presented a thorough theoretical background to our method based on structural causal models and Reichenbach's common cause principle.

A drawback of our method is that the researcher has to provide the feature of interest X prior to testing whether it is relevant. Our method does not propose features for testing. Nevertheless, in many situations either prior knowledge tells us which features might be relevant to a classifier, or we can use known methods to determine features as candidates for testing. One example of such methods are saliency maps. If we, for example, classify skin lesions, we can use a saliency map to highlight an area. If the skin lesion is highlighted in many images, we can use our method to understand which specific feature of the lesion is important. We can test the shape, the symmetry, the mean color, or others. In this way our method can be used in addition to saliency maps to understand a classifier further.

Additionally, we can evaluate the relevance of features that cannot be highlighted by saliency maps. In Section 4, we evaluate the position of the object as a feature. This feature cannot be evaluated using saliency maps.

As discussed in Section 2, our method will identify features as relevant, if they contain a relevant feature or if they are an invertible function of a relevant feature. This can lead to the method identifying multiple redundant versions of the same feature. To this end, additional feature to feature evaluations can be done using mutual information. We leave this for future work.

Whenever we apply our method, we have to select an independence test. Testing independence on finite samples is a hard problem in itself, and no single independence test is optimal for all kinds of data. When applying this method, we have to select an independence test carefully, depending on the data. For the experiments in this paper we used the RCOT [31]. To select a suitable independence test, we point the reader to [31] and [10] and the references therein.

In multiple experiments, we could demonstrate that the method can be used in relevant real-life situations. We demonstrated multiple experiments on MS COCO, CUB200, and HAM10000. In these experiments, we used the method to evaluate classifiers beyond there generalization performance on a holdout test set. We compared different classifiers beyond their generalization performance, and we evaluated whether a feature that is relevant to a human expert is relevant to a classifier. These additional evaluations can be used to generate trust in a classifier. They can help us to understand whether an existing classifier can be adapted to a new task, sharing some, but not all of the features of its old task. They can help to identify whether a neural network is biased or whether it bases its decision on meaningful features.

References

1. Adebayo, J., Gilmer, J., Goodfellow, I., Kim, B.: Local explanation methods for deep neural networks lack sensitivity to parameter values. arXiv preprint arXiv:1810.03307 (2018)
2. Barry-Jester, A.M., Casselman, B., Goldstein, D.: The new science of sentencing. The Marshall Project **4** (2015)
3. Berkeley, G.: A treatise concerning the principles of human knowledge. JB Lippincott & Company (1881)
4. Chen, Z.M., Wei, X.S., Wang, P., Guo, Y.: Multi-label image recognition with graph convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5177–5186 (2019)
5. Cui, Y., Song, Y., Sun, C., Howard, A., Belongie, S.: Large scale fine-grained categorization and domain-specific transfer learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4109–4118 (2018)
6. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3429–3437 (2017)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
8. Goyal, Y., Shalit, U., Kim, B.: Explaining classifiers with causal concept effect (CaCE). arXiv preprint arXiv:1907.07165 (2019)
9. Granger, C.W.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society pp. 424–438 (1969)
10. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: International conference on algorithmic learning theory. pp. 63–77. Springer (2005)
11. Jarisa, W., Henze, R., Kücükay, F., Schneider, F., Denzler, J., Hartmann, B.: Fusionskonzept zur reibwertschätzung auf basis von wetter- und fahrbahnzustandsinformationen. In: VDI-Fachtagung Reifen - Fahrwerk - Fahrbahn. pp. 169 – 188 (2019)
12. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). arXiv preprint arXiv:1711.11279 (2017)
13. Kindermans, P.J., Hooker, S., Adebayo, J., Alber, M., Schütt, K.T., Dähne, S., Erhan, D., Kim, B.: The (un) reliability of saliency methods. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 267–280. Springer (2019)
14. Kretschmer, M., Runge, J., Coumou, D.: Early prediction of extreme stratospheric polar vortex states based on causal precursors. Geophysical research letters **44**(16), 8592–8600 (2017)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
16. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking clever hans predictors and assessing what machines really learn. Nature communications **10**(1), 1–8 (2019)
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)

18. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
19. Mendonça, T., Ferreira, P.M., Marques, J.S., Marcal, A.R., Rozeira, J.: Ph 2-a dermoscopic image database for research and benchmarking. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 5437–5440. IEEE (2013)
20. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (2018)
21. Mopuri, K.R., Garg, U., Babu, R.V.: Cnn fixations: An unraveling approach to visualize the discriminative image regions. IEEE Transactions on Image Processing **28**(5), 2116–2125 (2018)
22. Pearl, J.: Causality. Cambridge university press (2009)
23. Perez, F., Vasconcelos, C., Avila, S., Valle, E.: Data augmentation for skin lesion analysis. In: OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis, pp. 303–311. Springer (2018)
24. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
25. Reichenbach, H.: The Direction of Time. Dover Publications (1956)
26. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat: Deep learning and process understanding for data-driven earth system science. Nature **566**(7743), 195–204 (2019)
27. Reimers, C., Requena-Mesa, C.: Deep learning - an opportunity and a challenge for geo- and astrophysics. In: Skoda, P., Adam, F. (eds.) Knowledge Discovery in Big Data from Astronomy and Earth Observation, chap. 13, pp. 251–266. Elsevier (2020)
28. Simon, M., Rodner, E.: Neural activation constellations: Unsupervised part model discovery with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1143–1151 (2015)
29. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
30. Stolz, W., Hölzel, D., Riemann, A., Abmayr, W., Przetak, C., Bilek, P., Landthaler, M., Braun-Falco, O.: Multivariate analysis of criteria given by dermatoscopy for the recognition of melanocytic lesions. In: Book of Abstracts, Fiftieth Meeting of the American Academy of Dermatology, Dallas, Tex: Dec. pp. 7–12 (1991)
31. Strobl, E.V., Zhang, K., Visweswaran, S.: Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. Journal of Causal Inference **7**(1) (2019)
32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
33. The Cornell Lab of Ornithology: All about birds. `https://www.allaboutbirds.org/guide/White-crowned_Sparrow/species-compare/64980371`, accessed: 2020-03-01
34. Tschandl, P., Rosendahl, C., Kittler, H.: The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data **5**, 180161 (2018)

35. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
36. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
37. Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with image-level supervisions for multi-label image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5513–5522 (2017)
38. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595 (2017)