Pattern Analysis and Applications (2022) 25:315–335
https://doi.org/10.1007/s10044-021-01055-y

THEORETICAL ADVANCES

Explainable image classification with evidence counterfactual

Tom Vermeire[1] · Dieter Brughmans[1] · Sofie Goethals[1] · Raphael Mazzine Barbossa de Oliveira[1] · David Martens[1]

Received: 20 May 2021 / Accepted: 13 December 2021 / Published online: 14 January 2022

Abstract

The complexity of state-of-the-art modeling techniques for image classification impedes the ability to explain model predictions in an interpretable way. A counterfactual explanation highlights the parts of an image which, when removed, would change the predicted class. Both legal scholars and data scientists are increasingly turning to counterfactual explanations as these provide a high degree of human interpretability, reveal what minimal information needs to be changed in order to come to a different prediction and do not require the prediction model to be disclosed. Our literature review shows that existing counterfactual methods for image classification have strong requirements regarding access to the training data and the model internals, which often are unrealistic. Therefore, SEDC is introduced as a model-agnostic instance-level explanation method for image classification that does not need access to the training data. As image classification tasks are typically multiclass problems, an additional contribution is the introduction of the SEDC-T method that allows specifying a target counterfactual class. These methods are experimentally tested on ImageNet data, and with concrete examples, we illustrate how the resulting explanations can give insights in model decisions. Moreover, SEDC is benchmarked against existing model-agnostic explanation methods, demonstrating stability of results, computational efficiency and the counterfactual nature of the explanations.

Keywords Image classification · Counterfactual explanation · Explainable artificial intelligence · Search algorithms

1 Introduction

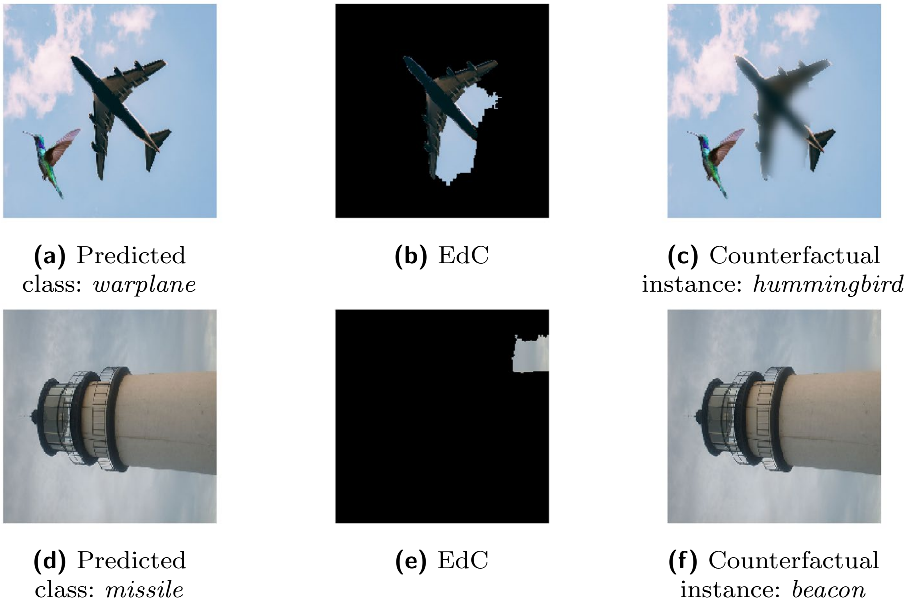
The use of advanced machine learning techniques for image classification has known substantial progress over the past years. The significant improvements in predictive performance, mainly due to the use of deep learning [33], have come at a cost of increased model complexity and opacity. As a result, state-of-the-art image classification models are used in a black-box way, without the ability to explain model decisions.

The need for explainability has become an important topic, generally referred to as explainable artificial intelligence (XAI) [2, 5, 21, 24, 38]. Often cited motivations are increased trust in the model, compliance with regulations and laws, and derivation of insights and guidance for model debugging [16, 21]. Additionally, the lack of explainability is considered a major barrier for the adoption of automated decision making by companies [5, 7, 11, 42]. As image classification for critical decisions is gaining ground, explainability is also becoming important in that context. We refer to applications such as medical image diagnosis [17], damage assessment in insurance [45] and self-driving cars [13], just to name a few, for which data subjects obviously demand an explanation. Explainability becomes even more important when severe misclassifications occur [34, 46]. Besides physical and/or reputational damage caused by the misclassification itself, companies should be able to explain what went wrong, as to prevent this from happening in the future and to restore trust.

For image classification, it can be argued that a good explanation allows to reveal an understandable and insightful pattern that led to the classification. If the pattern of a correct classification is true in the real world, this contributes to trust in the working of the model. Furthermore, a good explanation should show for misclassifications why the error was made to provide input for model improvement. Third, an explanation can also reveal that a correct classification has occurred for wrong reasons, which cannot be derived from the black-box prediction itself. Striking examples are the presence of snow to classify images as wolf [41] and the presence of a copyright tag to classify images as horse [31].

✉ Dieter Brughmans
Dieter.Brughmans@uantwerpen.be

[1] University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium


(a) Predicted class: *warplane*

(b) EdC

(c) Counterfactual instance: *hummingbird*

(d) Predicted class: *missile*

(e) EdC

(f) Counterfactual instance: *beacon*

Fig. 1 Examples of SEDC(-T) explanations: (i) removing the body of the *warplane* leads to the image being classified as *hummingbird* (see Fig. 1b), (ii) Removing clouds in the background resembling a *missile* exhaust plume leads to the misclassified image being correctly classified as *beacon* (see Fig. 1e)

In such settings, the discovered insights can also be used for model improvement.

Our literature review on explanations methods for image classification shows the increased interest in counterfactual explanations over feature importance rankings. As will be argued, current counterfactual explanation methods have however strong and sometimes infeasible requirements regarding access to model internals and training data. Building on previous work, we therefore propose a model-agnostic counterfactual explanation approach to explain individual image classifications, which only requires the image of interest and the black-box classification model (no need for access to the training data or model internals). The resulting explanations reveal the segments that must be removed (so-called evidence counterfactual or EdC) to change an image classification. The examples shown in Fig. 1 illustrate that the explanations can give insight in model decisions (blurring the plane part of the image changes the classification from *warplane* to *hummingbird*) and model failures (blurring the clouds that resemble exhaust plumes behind a missile, changes the misclassification from *missile* to the correct classification *beacon*). These will be detailed further in Sect. 4.

The contributions of this paper are threefold. First, we provide an overview of existing counterfactual explanation methods for image classification and propose a summarizing framework. Second, we introduce the novel model-agnostic methods, SEDC and SEDC-T, to generate instance-level explanations for image classification. Third, concrete examples illustrate how the resulting explanations can give insights in model decisions and large-scale experiments show the effectiveness and efficiency of our approach.

2 Related work

In general, an explanation provides an answer to a *why* question. In classification tasks, this question becomes: *why was the classification made?* Answering this question has ignited a whole research field [2, 5, 21, 24, 38]. In this section, we will discuss the main approaches that relate to our setting.

Multiple explanation methods for (image) classification have been proposed in the literature. A distinction can be made between global explanations, which apply to a model in general, and instance-level explanations, which focus on isolated model predictions [37]. In this paper, we focus on the latter. The main approaches to explain individual instances' predictions are feature importance and counterfactual methods, which will be discussed briefly next.

2.1 Feature importance methods

Feature importance methods provide a ranked list of the features that are deemed most important for the prediction made on that instance. For an image explanation, this corresponds to showing the parts of the image (pixels or segments) that have contributed the most to the prediction. LIME [41] and the model-agnostic implementation of SHAP [36] are popular feature importance methods on the instance-level that provide a set of features (segments) with the coefficients of a linear model that is created around the instance that needs to be explained, and as such approximates the predictions of the actual prediction model around that point. Although these methods demonstrate the contribution of each feature to the overall prediction, they do not use the decision boundary (and hence provide no counterfactual), thereby losing the advantage to understand what needs to be changed in order to receive a desired outcome. Other drawbacks include the number of features in an explanation that needs to be set by the user, and the existence of a randomization component in the method (the generation of random data points around the instance to be explained), which leads to unstable results [4]: running the explanation method for a given instance and a given prediction model twice can lead to two different explanations. It also does not make the influence of interactions between features clear [19], as it uses a linear approximation. Finally, the computational time to generate explanations can be very large: for example, Lapuschkin [30] reports around 10 minutes of computation time needed to generate a LIME explanation for a single prediction of the GoogleNet image classifier. That being said, they do offer valuable insights into an individual prediction (as demonstrated by their popularity) and are model-agnostic.

Other feature importance methods can be used to create visual heat maps on top of the pixels. Occlusion is a first general strategy, that measures the influence of each pixel, by masking regions and assessing the impact on the output score [52, 53]. A second approach is taken by Bach et al. [6] who introduce Layer-Wise Relevance Propagation (LRP) as a model-specific method to create instance-level explanations for neural networks. A third approach calculates the gradient of the prediction function at the instance to be explained, which indicates the importance of each pixel/feature in the prediction score [44]. The latter two approaches require access to the model weights and can therefore not be used when the prediction model is only available as a scoring function, without access to the model internals. Additionally, an important disadvantage of pixel-wise heat map methods is the low abstraction level of the explanations [42]. Since individual pixels are meaningless for humans, it is not always straightforward to derive interpretable concepts from it.

In general, a larger issue with feature importance methods is what they actually explain. Fernandez et al. [19] argue that feature importance rankings do not explain a classification, but rather a *prediction score*. End users typically wish to understand why a certain impactful *decision* has been made. And while data scientists often focus on prediction scores (cf. the popularity of the ROC and AUC), they as well wish to understand certain classifications (instead of prediction scores) to answer the question of *why was this image misclassified?* That brings us to counterfactual explanations, which explain a *classification* made on a data instance, by a prediction model.[1]

2.2 Counterfactual reasoning for image classification

Many authors in the field of philosophy and cognitive science have raised the importance of contrastive explanations [35, 38]. Martens and Provost [37] were the first to apply this idea for predictive modeling, in the context of document classification, and have sparked a large set of novel counterfactual methods to be introduced [9, 10, 12, 19, 29, 40, 51]. Apart from the contrastiveness, counterfactual explanations have other benefits. It is argued that they are more likely to comply with recent regulatory developments such as GDPR. Wachter et al. [51] state that counterfactual explanations are well-suited to fill three important needs of data subjects: information on how a decision was reached, grounds to contest adverse decisions and an idea of what could be changed to receive a desired outcome. Moreover, formulating an explanation as a set of features does not put constraints on model type and complexity [7], which should make it robust for developments in modeling techniques. Finally, the explanation can be done without disclosing the entire model [7], which allows companies to give only the necessary information without revealing trade secrets.

Several authors have used approaches that are closely related to counterfactual reasoning for image classification. Adversarial example methods for example, which aim at finding very small image perturbations that lead to false classifications [22, 47, 48]. This has proven useful to protect a classifier against attempts to deceive it. However, since the found perturbations are often too small to be visible for humans (in extreme cases only one pixel), they cannot be used as an interpretable counterfactual explanation.

Other authors explicitly used counterfactual explanations for explainability in image classification. These papers are summarized in Table 1 in terms of the following dimensions. We also include our novel approach, SEDC(-T), which will be further described in Sect. 3.

1. Model-agnostic (MA): does method work without access to model internals?
2. Training data-agnostic (TA): does method work without access to training data?
3. Abstraction-level (AL): what is the granularity of the features?
4. Addition of evidence (AE): is purposefully adding evidence allowed?
5. Explanation focus (EF): does the explanation focus on the changes or the counterfactual?

The following four important observations can be made from the literature overview. First, there is quite some ambiguity and vagueness regarding the terminology used in the

[1] Fernandez et al. [19] additionally demonstrate that the feature importance rankings are not necessary nor sufficient to be included in a counterfactual.

Table 1 Summary of counterfactual explanation methods: reported dimensions are whether method is model-agnostic (MA), whether method is training data-agnostic (TA), the abstraction-level (AL), whether adding evidence is allowed (AE) and whether the explanation focuses on the changes or the resulting counterfactual (EF)

Method	MA	TA	AL	AE	EF
Dhurandhar et al. [15]	No	No	Pixel	Yes	Changes
Van Looveren et al. [49]	No	No	Pixel	Yes	Counterfactual
Joshi et al. [28]	No/Yes	No	Pixel	Yes	Counterfactual
Hendricks et al. [26]	No	No	Conceptual (textual)	Yes	Counterfactual
Goyal et al. [23]	No	No	Conceptual	Yes	Counterfactual
Akula et al. [3]	No	No	Conceptual	No	Changes
SEDC(-T)	Yes	Yes	Conceptual	No	Changes

context of counterfactual explanations for image classification. Initial work on counterfactual explanations focuses on representing the *changes* that must be applied to alter a classification. Because it was first used for models based on textual and traditional data, an important advantage is reducing the typically large feature space to a smaller and more interpretable set of features. In the context of explaining image classifications, however, some authors use the modified (counterfactual) image as an explanation [23, 28, 49], while others focus on representing the necessary changes between the instance to be explained and the counterfactual [3, 26]. A possible reason for only considering the counterfactual as an explanation is that the necessary changes themselves are not interpretable. For instance, only showing the pixels that change the classification of digits [28, 49] or only showing a part of an image belonging to the counterfactual class [23] would clearly not suffice as an interpretable explanation. Since all authors refer to generating counterfactual explanations, we want to make a clear distinction between the necessary *changes* to alter an image classification (the evidence counterfactual or EdC, which we pronounce as 'Ed See') and the *counterfactual* image resulting from these changes (counterfactual). Our approach aims at finding an explanation that identifies the changes that are necessary to alter a classification.

Second, pixel-level explanations have their merits in toy examples, for example when classifying digits, but quickly lose their interpretability in real-life applications. This might be a reason why these pixel-explanation methods are typically tested on relatively simple datasets and classification tasks, such as MNIST [32]. However, we see a shift towards explanations at a higher abstraction level. In line with this, we use image segments to compose a conceptual counterfactual explanation and test our approach on a broad classification task and dataset (sample of ImageNet data [27]).

Third, existing counterfactual approaches allow the purposeful addition of evidence to the image, e.g., adding parts of an image belonging to the counterfactual class [23] or adding concepts to the image supporting the counterfactual class [3]. This leads to an EdC containing evidence that is actually not present in the original image, which seems rather counter-intuitive in the context of images. It can also be argued that this does not necessarily lead to interpretable explanations (e.g., is mixing two types of animals in one image semantically clear?) or that the explanation is not necessarily useful (e.g., any image can be turned into a zebra prediction by simply adding a zebra to the image). Therefore, we limit our search for explanations to the removal of evidence, which results in EdCs only containing evidence that is present in the image of interest.

Fourth, there is the requirement for most methods to have access to both the model internals (as most methods are not model-agnostic) and the training data. In practical applications, this is often not feasible. Many companies use classification models built by external vendors, e.g., Google Cloud Vision,[2] Amazon Rekognition[3] and the Computer Vision service in Microsoft Azure.[4] Even if the model itself would be open source (which is often not the case), the training data are rarely available, as this is either too large to efficiently share, or considered part of the vendor's proprietary assets. This implies that in these cases, the previously proposed counterfactual methods cannot be used. Moreover, model-agnostic methods have a wide(r) applicability as they are not limited to specific model types or architectures. From an academic point of view, such approaches are arguably also more likely (or at least easier) to be re-used and built upon by other researchers. Our approach aims to fill this important gap in the literature by proposing the first model-agnostic counterfactual explanation method for image classification, only based on the black-box model and the image of interest.

[2] https://cloud.google.com/vision.

[3] https://aws.amazon.com/rekognition/.

[4] https://azure.microsoft.com/nl-nl/services/cognitive-services/computer-vision/.

3 Methodology

Martens and Provost [37] introduce a model-agnostic search algorithm (SEDC) to find the EdC for document classifications. The EdC explanation is an irreducible set of features (i.e., words) that, in case they were not present, would alter the document classification. In this context, irreducible means that removing any subset of the EdC would not change the classification. We explore how an adapted version of this method can be used to generate visual counterfactual explanations for image classification. In the remainder of this paper, we will refer to this algorithm as Search for EviDence Counterfactual (SEDC[5]).

Consider an image I assigned to class c by a classifier C_M. The objective is to find an EdC of the following form: an irreducible set of segments that leads to another classification after removal. The segmentation and removal of segments will be discussed in Sect. 3.3.

EdC can be formally defined as a set of evidence E (segments in the image) for which applies:

$$E \subseteq I \text{ (segments in image)} \tag{1}$$

$$C_M(I \backslash E) \neq c \text{ (class change)} \tag{2}$$

$$\forall E' \subset E : C_M(I \backslash E') = c \text{ (irreducible)} \tag{3}$$

Remember that the EdC points at the changes that are necessary to alter a classification, while the counterfactual is the result of these changes.

3.1 SEDC for image classification

In electronic format, an image is a collection of pixel values (one value per pixel for grayscale images, three values (RGB) per pixel for colored images). These individual pixel values are typically used as input features for an image classifier. As interpretable concepts in images are embodied by groups of pixels or segments, we propose to perform a segmentation, similarly to LIME [41] and SHAP [36]. In line with the reasoning behind SEDC for document classification, the goal is to find a small set of segments that would, in case of not being present, alter the image classification. The original SEDC-algorithm [37] was applied to binary document classifications models outputting a single prediction score reflecting the probability of belonging to the class of interest. In image classification applications, one is often confronted with more than two possible categories, each with its own prediction score. Therefore, we generalize SEDC by enabling the occurrence of multiclass problems. More specifically, additional segments are selected by looking for the highest reduction in predicted class score.

A short version of the pseudo-code is outlined in Algorithm 1. A more detailed version is outlined in Algorithm 2 (A). SEDC takes an image of interest, an image classifier with corresponding scoring function and a segmentation as inputs and produces a set of EdCs as output. Each individual EdC is a set of segments that leads to a class change after replacement. A heuristic best-first search is performed in order to avoid a complete search through all possible segment combinations. The best-first is each time selected based on the highest reduction in predicted class score, and subsequently, all expansions with one additional segment are considered. This search continues until one or more same-sized EdCs are found after an expansion loop (i.e., the set of explanations is not empty).

In the procedure outlined above, already explored combinations remain part of the considered combinations to expand on. As a consequence, it is possible that, when searching for the best-first, the algorithm returns to a smaller combination in the search tree (for instance, after expanding combinations with three segments, the best-first might again be a combination of two segments). To assert that the algorithm does not get stuck in an endless loop by repeatedly returning to the same combination in the search tree, a selected combination is each time removed after all expansions with one additional segment are created. We refer to this as the pruning step.

[5] We pronounce this as ‘Sed See’.

Algorithm 1 SEDC

```
Inputs:
$I$ % Image to classify
$C_M : I \rightarrow \{1, 2, ..., k\}$ % Trained classifier with $k$ classes
$S = \{s_i, i = 1, 2, ..., l\}$ % Segmentation of the image with $l$ segments

Procedure:
$R = \{\}$ % List of EdCs
for $s_i$ in $S$ do
    if class change after removing $s_i$ from $I$ then
        $R = R \cup \{s_i\}$
    end if
end for
while $R = \emptyset$ do
    Select $best$ % Best-first: segment set with highest reduction in predicted class score
    $best\_set$ = all expansions of $best$ with one segment
    for $C_0$ in $best\_set$ do
        if class change after removing $C_0$ from $I$ then
            $R = R \cup \{C_0\}$
        end if
    end for
end while

Output:
EdCs in $R$
```

The final set *R* can consist of one or more EdCs, depending on how many expansions of the last combination result in a class change. In that case, we select the EdC with the highest reduction in predicted class score. Theoretically, it is possible that no class change occurs and, consequently, the while loop in the algorithm never ends. To prevent this from happening, one or more additional conditions could be added to this while loop (e.g., maximum number of iterations, maximum computation time, etc.).

As shown by Martens and Provost [37], SEDC automatically results in irreducible explanations for linear classification models. This irreducibility cannot be guaranteed when using a nonlinear model, as is generally the case for image classification. Therefore, an additional local search can be performed by considering all possible subsets of the obtained EdC. However, as argued in C, the irreducibility condition is rarely violated, and the additional search can be considered an unnecessary and time-consuming step.

3.2 SEDC with target counterfactual class

As image classification is often a multiclass problem, it is useful to generate counterfactual explanations for which the counterfactual class is predefined (not just any other class). For this purpose, we propose an alternative version SEDC-T in which segments are iteratively removed until a predefined target class is reached. A detailed version is outlined in Algorithm 2 (A). The target class serves as an additional input parameter, and segments are selected based on the largest difference between the target class score and the predicted class score. In case more than one EdC is found, the EdC leading to the highest increase in target class score can be selected. Again, one or more additional conditions could be added to prevent the occurrence of an infinite while loop in case the target class is never reached.

SEDC-T allows for the generation of more nuanced explanations, since one can find out why the model predicts a class over another class of interest. This can certainly be useful for explaining misclassifications. In that case, it might be relevant to know why an image is not assigned to the correct class, rather than to know why it is assigned to the incorrect class (e.g., why is Fig. 1d not classified as *beacon*?). Opposed to linear models for binary cases, these two notions are not the same in multiclass problems with nonlinear models. Therefore, by generating counterfactual explanations with the correct class as target counterfactual class, it is possible to identify those parts of the image that led to its misclassification.

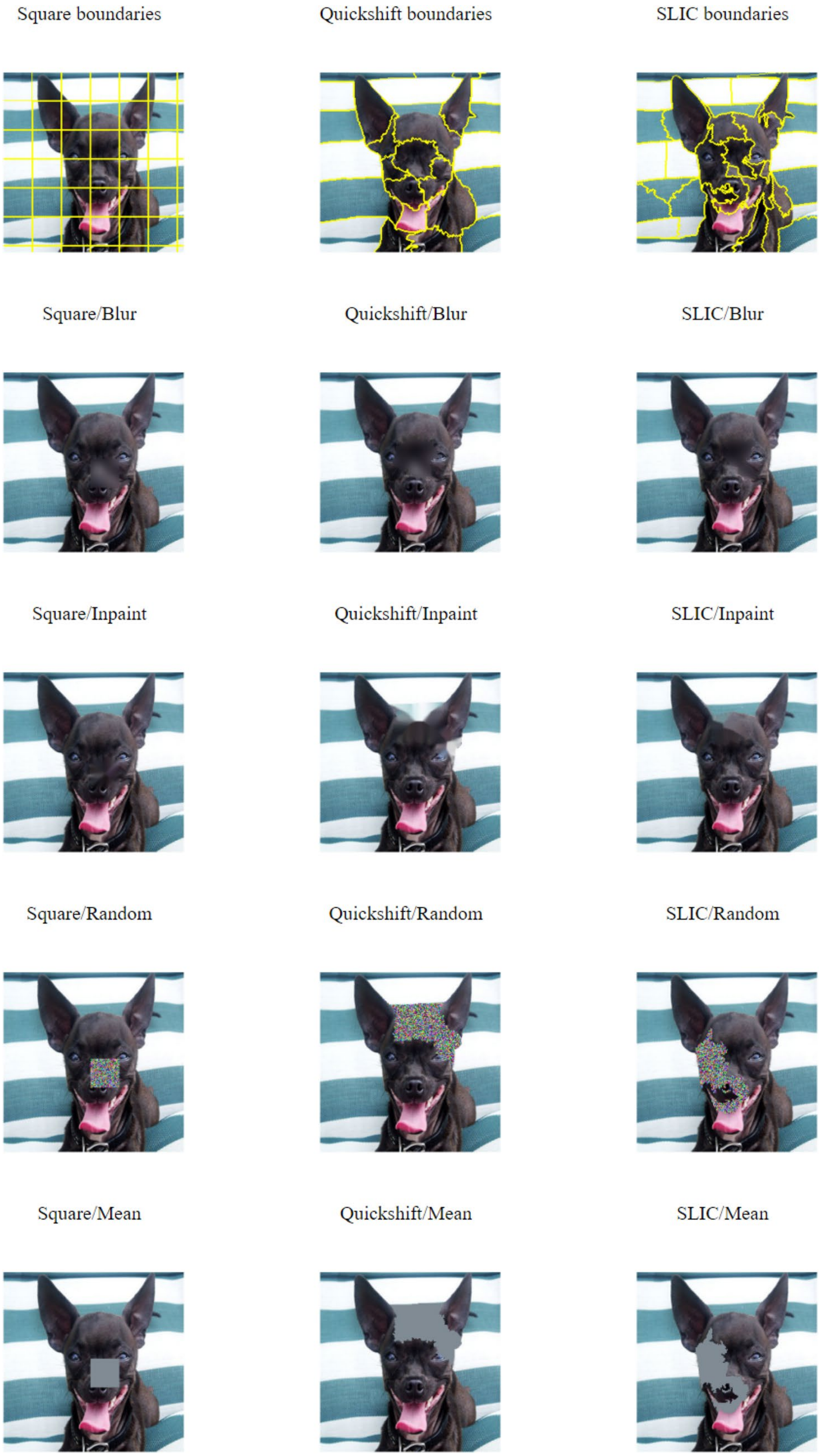
3.3 Building blocks

Two important building blocks of our approach are the segmentation and the segment replacement method. Both building blocks must be deliberately decided upon for an actual implementation and application of the algorithm.

First, the segmentation can take different forms. For instance, one can choose a (naive) squared segmentation by dividing the image in squares of equal size. However, the meaning of the resulting segments is highly dependent on the specific image. Therefore, a more suited approach is using an advanced segmentation algorithm that uses the numerical color values to obtain a meaningful grouping in segments (e.g., quick shift [50], SLIC [1], graph-based segmentation [18] and others).

Second, the removal of evidence is not straightforward in the context of image data. Each time one or more segments are removed, an image perturbation is created. For textual

Fig. 2 All combinations of segmentation and replacement methods illustrated on an image of a chihuahua



and behavioral data, generating an instance wherein certain features are not present is usually done by replacing the feature values by zeros. Setting the values of (groups of) pixels to zero in images corresponds to altering the color of the pixels to black (both for grayscale and colored images). For images wherein black has a strong presence and/or meaning, this might be problematic since replacing the color by black has no or little impact. The same applies for any other color chosen ex ante. Alternatively, the segment replacement can be based on calculated pixel values. For example, a segment can be changed to the mean/mode pixel values of the image as a whole, the segment itself or the neighboring segments. Also, more advanced imputation methods for images are possible (e.g., image inpainting or blurring).

With the aim to investigate the impact of using different building blocks, we performed experiments with three segmentation methods (uniform squares, quick-shift [50] and SLIC [1]) and four replacement methods (blurring [25], inpaint [8], mean and random). However, it must be highlighted that it's difficult to quantify the performance of different methods. One could, for example, look at the area size of each explanation. However, this tells you nothing about how clear the explanation is to an end-user. Instead of purely quantitative metrics, a more subjective (qualitative) evaluation process would include comparing how well the segmentation method matches the human perception of the parts and how extreme they perceive the replacement method. Therefore, results that are considered better in a quantitative perspective, may not be (necessarily) the best in qualitative evaluations.

In Fig. 2, we show an example of explanations for all combinations of segmentation and replacement methods. In the first row, the boundaries of the segmentation methods are shown. We argue that quickshift best matches the human perception of segments on these pictures. Although it's not perfect, there is a clear segment for, the tongue, forehead, nose, muzzle and each ear of the chihuahua. This is less so for SLIC where some segments contain both parts of the dog and background while others contain different body parts of the dog. For the squared segmentation method, there is an even worse semantic value in the segments of the image.

When comparing the replacement methods, we readily notice that both the random and mean methods drastically change the image. This is evident when analyzing the different explanations using squared segmentation of Fig. 2. While the Blur and Inpaint approaches make subtle adjustments to the original picture, both in terms of colors and shapes.

Based on this qualitative analysis and considering the counterfactual explanation premise that changes must be minimum, we decided to perform all our experiments using quickshift as our segmentation model and blurring as our replacement method, since those methods give, respectively, a semantic meaning to the changes and modifying as little as possible the original image. However, should better methods be developed in future, they can easily be integrated into both algorithms.

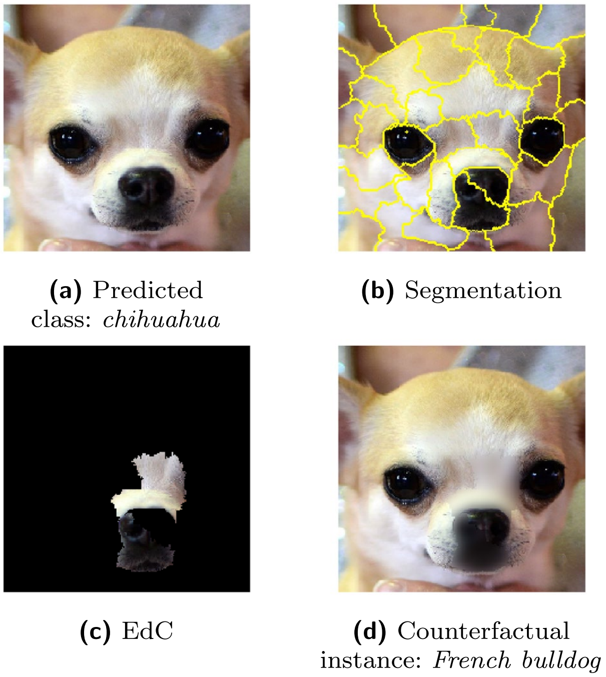

(a) Predicted class: *chihuahua*
(b) Segmentation
(c) EdC
(d) Counterfactual instance: *French bulldog*

Fig. 3 SEDC-T applied to *chihuahua* image

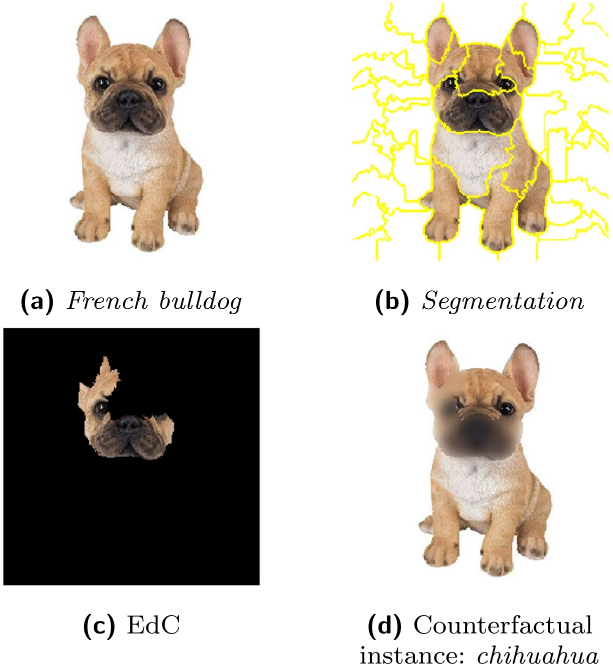


Fig. 4 SEDC-T applied to *French bulldog* image

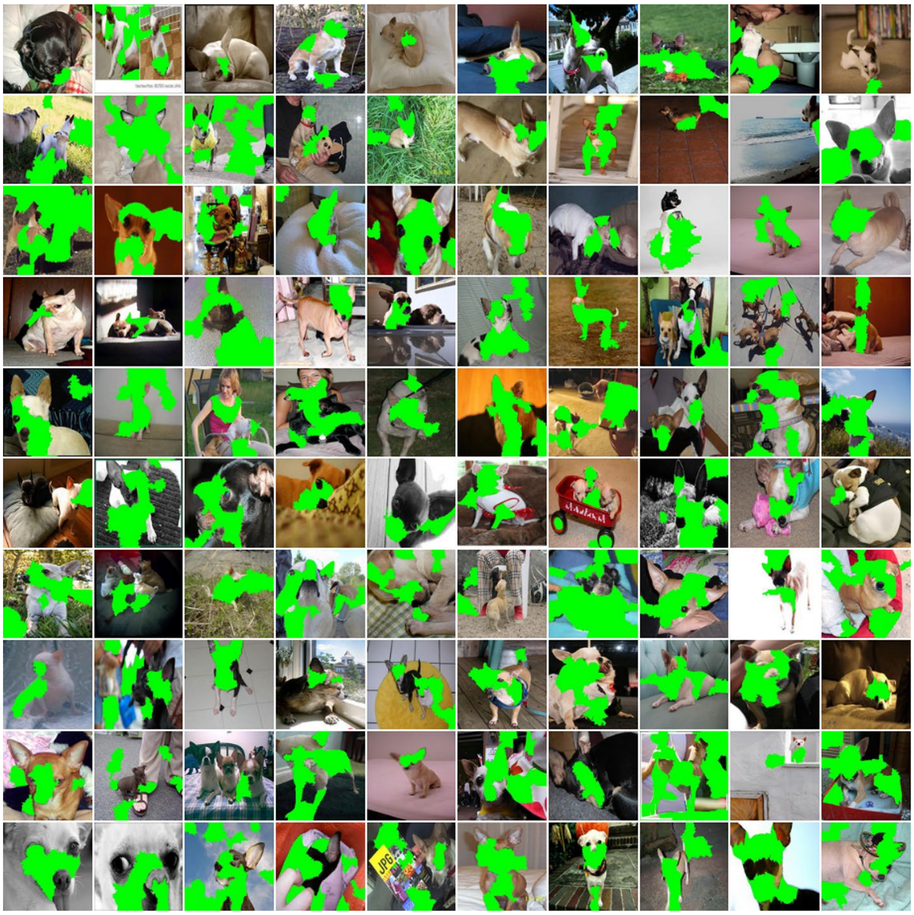


Fig. 5 Gallery highlighting the counterfactual regions for the factual image *Chihuahua* targeting the class *French bulldog*. Blurred segments are highlighted in green to improve visibility. The original gallery with blurred segments is shown in Fig. 12

4 Results

In this section, the results of experiments with SEDC and SEDC-T for image classification are discussed. First, the details of our approach are illustrated with a detailed example on a *chihuahua* image. Second, we illustrate how our approach can lead to insight in model decisions and how SEDC-T can be used to compare different counterfactual classes. Third, SEDC(-T) explanations are benchmarked against the existing model-agnostic explanation methods LIME, SHAP and occlusion.

For our experiments, SEDC and SEDC-T are implemented in *Python*.[6] Unless stated otherwise, we use quick shift segmentation [50] and blur segment removal (Gaussian smoothing [25]). In the benchmark, we use the respective available implementations of LIME[7] and SHAP.[8] Google's pre-trained MobileNet V2 model [43] is used as image classification model, as it is considered a fast, state-of-the-art classifier for regular personal devices. Since this model gives prediction scores for 1001 different categories, the highest scoring class is selected as predicted class. Image data are downloaded from ImageNet [27] (29,275 images in total with 20 different labels). Our experiments are conducted on a laptop with Intel i7-8665U CPU (1.90 Ghz) and 16 GB RAM.

4.1 Running example

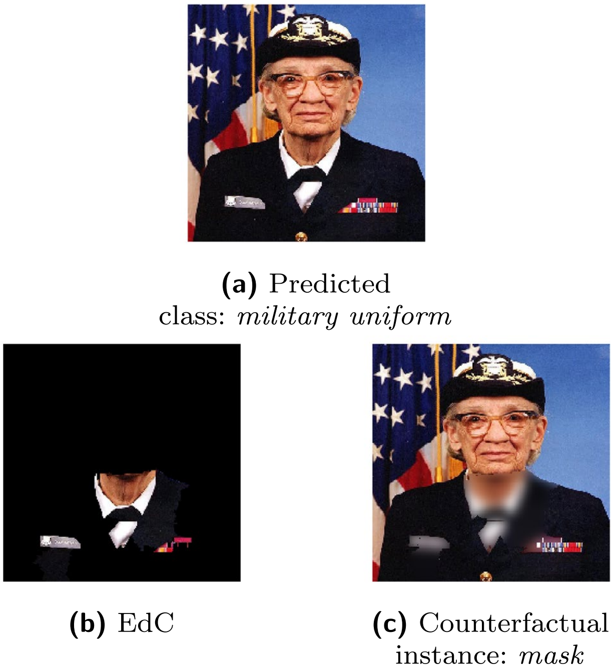
To clarify the working of SEDC(-T), we consider an arbitrary image of a *chihuahua* (see Fig. 3a), which is correctly classified by the model. We apply SEDC-T with blur segment replacement and set the second highest scoring class *French bulldog* as the target. Segments are shown in Fig. 3b. The resulting EdC is shown in Fig. 3c. Figure 3d contains the counterfactual instance leading to a class change. After replacing segments near the nose of the *chihuahua*, the image is classified as *French bulldog*.

It is interesting to compare the EdC and counterfactual instance with an image of an actual *French bulldog* (see Fig. 4a). At first glance, someone would probably identify the eyes of this *chihuahua* as the most important segments. However, the eyes are not that different from the eyes of a

[6] The Python code will be made publicly available on Github upon acceptance of the paper and can be made available to the reviewers if needed.

[7] https://github.com/marcotcr/lime.

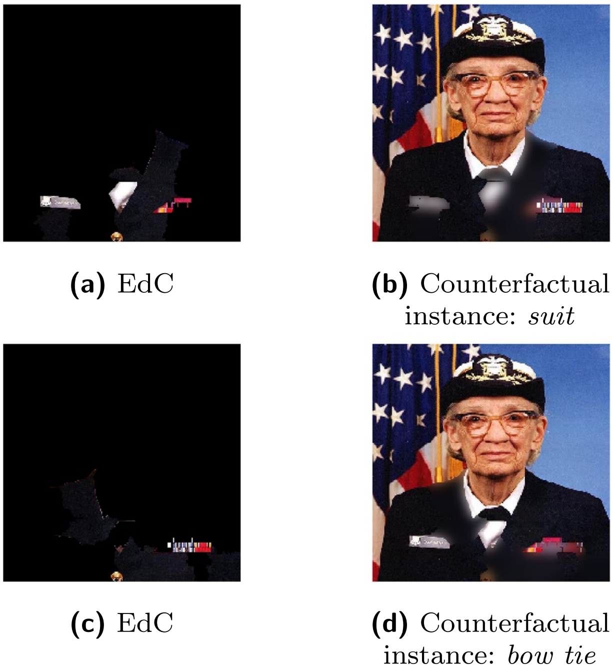
[8] https://github.com/slundberg/shap.


(a) Predicted class: *military uniform*

(b) EdC

(c) Counterfactual instance: *mask*

Fig. 6 SEDC applied to *military uniform* image: why military uniform and not any other class?


(a) EdC

(b) Counterfactual instance: *suit*

(c) EdC

(d) Counterfactual instance: *bow tie*

Fig. 7 SEDC-T applied to *military uniform* image: why military uniform and not **b** suit, or **d** bow tie?

French bulldog. We also verify whether SEDC(-T) is consistent in highlighting similar segments in different images for the same image class. This consistency can be seen in Fig. 5, where consistently the muzzle and nose are the selected regions to be replaced with blurred segments, for a random selection of 10×10 images of chihuahuas.[9] It's important to observe that in this figure the EdC explanations are indicated in green instead of blurred regions, as to make them more visible, the original counterfactual outputs were included in "Appendix D: SEDC-T Experiment—blurred images".

In addition, we apply SEDC-T to the *chihuahua* example, this time in the opposite direction, to answer the question: "Why is this image classified as a *French bulldog* and not as a *chihuahua*?" The *French bulldog* image is taken as the input image, and *chihuahua* is taken as target counterfactual class. This results in the EdC and counterfactual instance in Fig. 4c, d. Also here, the nose of the dog is considered the most distinguishing characteristic, supporting the previous explanation.

When using other segmentation methods (squared and SLIC [1]), the resulting explanations point at the nose of the *chihuahua* as well. Also, SEDC(-T) with other segment replacement methods (segment inpainting [8] and segment blurring by applying Gaussian smoothing [25]) consistently point to the nose of the *chihuahua* as part of the EdC.

Table 2 Experiment: SEDC-T applied to misclassifications

# images	Target found	Target not found
2121	1831 (86%)	290 (14%)

4.2 SEDC(-T) for insight in model decisions

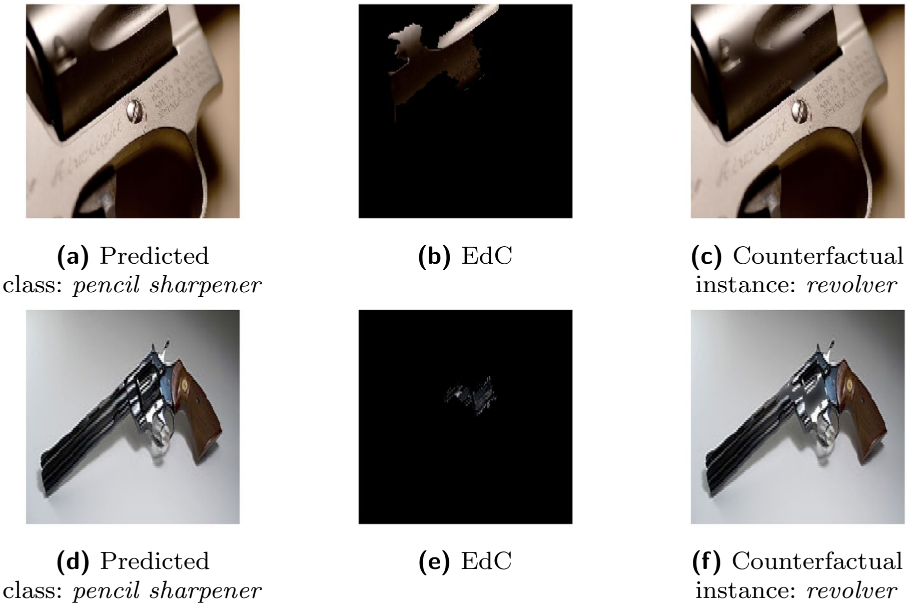
4.2.1 Insight in correct classifications

The *warplane* and *chihuahua* examples illustrate how SEDC can create explanations that give more insights in the model decision. These can help to better understand the model decision and assess its quality. By identifying the discriminative segments of the image, SEDC shows the regions that matter for the warplane classification and reveals what, according to the model, differentiates the *chihuahua* from a *French bulldog*.

As a next example, we consider an image that is classified as *military uniform* shown in Fig. 6a. Applying SEDC results in the EdC and counterfactual instance shown in Fig. 6b, c. The explanation points to segments containing the name tag, the medal ribbons, the bow tie and the neck of the person. After blurring these parts, the image is classified as *mask*.

[9] This grid size is simply chosen based on how well it looks on a single page.

Fig. 8 SEDC-T applied to *revolver* classified as *pencil sharpener*


(a) Predicted class: *pencil sharpener*
(b) EdC
(c) Counterfactual instance: *revolver*
(d) Predicted class: *pencil sharpener*
(e) EdC
(f) Counterfactual instance: *revolver*

One could also wonder what is needed to classify the image as *suit* instead of *military uniform* and apply SEDC-T with *suit* as target counterfactual class. This results in the EdC and counterfactual instance shown in Fig. 7a, b. The explanation also points to the importance of the name tag, the medal ribbons and the buttons for classifying the image as *military uniform* over *bow tie*, while the neck is in this case not part of the EdC.

We also apply SEDC-T with *bow tie* as target and obtain the EdC and counterfactual instance shown in Fig. 7c, d. Here, segments with the medal ribbons and the button are removed, but the ones with the bow tie are kept in place. The different explanations contain segments with elements that are, according to the model, distinctive for a *military uniform* over several other classes. Hence, the use of SEDC-T allows to get insight in the decision boundaries between the different classes and to assess whether the working of the model can be understood.

Fig. 9 SEDC-T applied to *mouse* classified as *soccer ball*

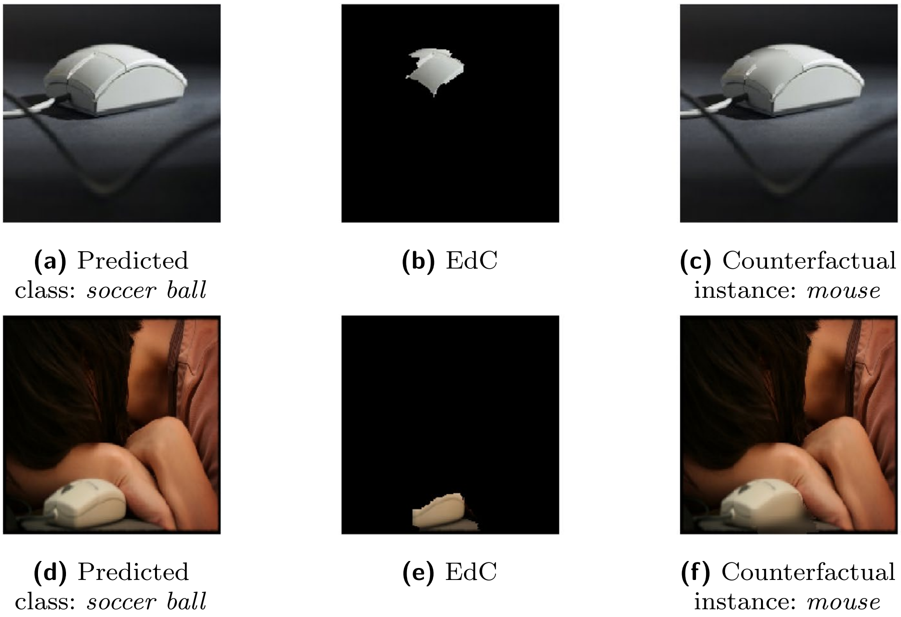

(a) Predicted class: *soccer ball*
(b) EdC
(c) Counterfactual instance: *mouse*
(d) Predicted class: *soccer ball*
(e) EdC
(f) Counterfactual instance: *mouse*

Table 3 Main cases where SEDC-T failed for misclassifications within time limit

Class	Correct class	# misclassified	# target not found
Desk	Mouse	57	36 (63%)
Desktop computer	Mouse	65	27 (42%)
Bearskin	Military uniform	33	13 (39%)

4.2.2 Insight in misclassifications for model improvement

In Sect. 1, we argued that model improvement is an important explainability objective. If it is possible to explain model errors (misclassifications), the explanation(s) can provide input to better understand why the model failed. This information could then be used for model debugging (e.g., by gathering additional training data focusing on the identified relevant parts). In the next experiment, we evaluate the effectiveness of SEDC-T in finding explanations for misclassifications.

For each of the 20 labels in our ImageNet data set, we verify which images are misclassified by our model and select the images belonging to the top five most occurring misclassifications per label. To these misclassified images (2121), SEDC-T is applied with the correct class as target and a maximum search time per image of 15 s. The results are summarized in Table 2. In 86% of the cases, SEDC-T finds an EdC leading to the correct class change. A detailed overview of the data and the results per class is given in Table 5 in F.

A closer look to the explanations led to some interesting observations. A first example is the misclassified *beacon* shown in Fig. 1e, where the EdC points to the background with clouds resembling an exhaust plume of a *missile*. Other examples can be found in Figs. 8 and 9. The EdCs in Fig. 8b, e identify the cylinder of the *revolver* as the reason for being mistaken for a *pencil sharpener*. Figure 9b, e point to a part of the *mouse* that is presumably confused with the bounded faces on a *soccer ball*. These examples show that our approach can point at biases in the training data that are learned by the classification model. By improving the quality of the training set and mitigating these biases (e.g., by providing more images of revolvers focusing on the cylinder), the classification model can be debugged.

For 290 images (14%), SEDC-T did not reach the correct class within the time limit of 15 s. In 289 of the 290 cases, SEDC-T does however result in a perturbation with an improved difference between the target and predicted class score after 15 s of search time. Even though the correct class change is not (yet) reached, it is thus almost always feasible to find perturbations that lie closer to the correct class.

We further investigate the cases where SEDC-T fails to find an explanation. In Table 3, the misclassifications are shown for which SEDC-T is successful in less than 70% of the cases and with an occurrence of at least 10.

SEDC-T also can have relevance when used to evaluate images with multiple objects where the final classification is an unwanted part of the image, such usage is further described in "Appendix E: SEDC-T for model insight in missclassifications—image with multiple objects".

4.3 Comparison with existing methods

Next, we compare our novel approach with existing methods. First, we illustrate the issues of feature importance ranking methods mentioned in Sect. 2.1 and compare such explanations with our counterfactual explanations. Second, we quantitatively benchmark our approach against existing model-agnostic explanation methods.

4.3.1 Limitations of feature importance explanation methods

As a first example of feature importance methods, LRP heat maps [6] for the *chihuahua* image are shown in Fig. 10. They reveal some drawbacks compared to the counterfactual explanations.

First, the heat maps do not entail an explanation size. By coloring pixels according to the implied feature importance

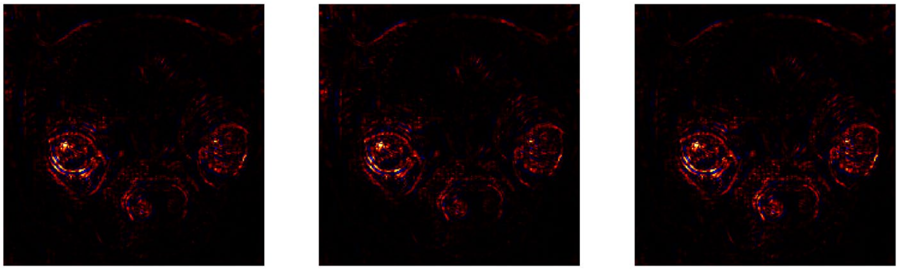


Fig. 10 LRP applied to *chihuahua* image

(a) LRP heat map for *chihuahua* **(b)** LRP heat map for *French bulldog* **(c)** LRP heat map for *tennis ball*

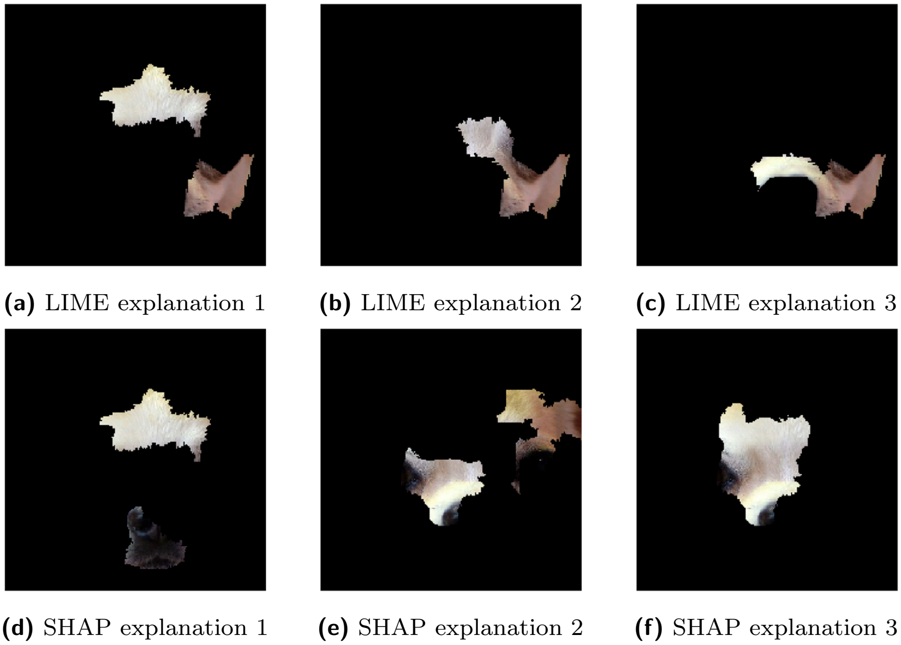

(a) LIME explanation 1
(b) LIME explanation 2
(c) LIME explanation 3
(d) SHAP explanation 1
(e) SHAP explanation 2
(f) SHAP explanation 3

Fig. 11 LIME and SHAP applied to *chihuahua* image

ranking, the user can get an idea of regions leading to the classification. However, this type of explanation does not tell what is minimally needed. The heat map does not allow to unambiguously assess whether the eyes are sufficient, or also the nose and contours of the head are needed. In contrast, SEDC automatically limits the size of the EdC to the parts that would alter the classification. This is considered useful, since the necessary size of an interpretable explanation can vary considerably between images.

Second, a heat map does not account for possible interdependence between features. Figure 10a points to the importance of the eyes and the nose for the classification. Though, it does not tell whether the nose would also be important in case the eyes were not present. SEDC takes these possible dynamics into account by reevaluating the importance of the image segments after every removal.

Third, the heat map explanations only give evidence supporting one class and are thus not contrastive in nature. Although it is possible to create heat maps for different classes and compare them, this is not always useful. Consider for instance the heat maps of the *chihuahua* image for the classes *French bulldog* and *tennis ball*, respectively, shown in Fig. 10b, c. They can hardly be distinguished from the *chihuahua* heat map and thus imply that the same image regions are important for all three classes (even for the *tennis ball class*). Therefore, it is impossible to derive why the image is classified as *chihuahua* over *French bulldog* or *tennis ball*. By contrast, SEDC bases its explanations on a class change and therefore searches for discriminative features (e.g., classified as *chihuahua* over *French bulldog* due to the nose).

We also generate LIME and SHAP explanations for the *chihuahua* image by making use of the same segmentation. A sample size of 1000 image perturbations is used for both methods.[10] Since applying SEDC results in an EdC consisting of two segments, only the two most important segments are shown for each method. Remember that this is a parameter that a user needs to set for LIME and SHAP. Taking into account possible explanation instability due to the sampling process, the explanation generation process was repeated three times. The LIME and SHAP explanations are shown in Fig. 11.

The fact that for both methods the three explanations differ for a fixed image and prediction model, illustrates the instability problem. A segment with a part of the *chihuahua*'s cheek is part of each LIME explanation, while the second segment shows another characteristic of the dog. The first SHAP explanation points to the importance of the *chihuahua*'s nose, the second contains a part of the eye, and the third shifts the focus toward the forehead. In contrast, SEDC is deterministic and always results in the same explanation.

The mentioned limitations of feature importance methods do also apply to LIME and SHAP: they do not provide an optimized explanation size (we took the size of SEDC explanations), the relative ordering of segments does not account for dependence between them, and the explanations are related to one class. Regarding the latter, we revisit the idea of comparing explanations for different classes. First, it is not guaranteed that the explanations are sufficiently different

[10] This sample size is used in LIME and SHAP tutorials for image classification.

Table 4 Benchmarking results of SEDC, LIME, SHAP & occlusion explanations: stability (%), median, mean and standard deviation of computation times (s), counterfactual nature (%) and counterfactual nature when explanation consists of multiple segments (%)

Criterion		SEDC	LIME	SHAP	occlusion
Stability (%)		**100**	61.13	53.01	**100**
Computation time (s)	Median	3.62	27.15	33.38	**1.83**
	μ	6.57	27.38	33.45	**1.85**
	σ	6.49	0.94	2.06	**0.31**
Counterfactual (%)		**100**	29.10	44.75	35.50
Counterfactual > 1 (%)		**100**	25.67	34.49	15.75

The best performing method for each metric is indicated in bold

to reveal discriminative regions. Moreover, the instability issue adds a layer of complexity, since the explanation for each of the classes is subject to chance. This implies that one should compare multiple versions of explanations for each of the classes, which obviously renders the process more difficult and uncertain.

4.3.2 Benchmark against model-agnostic explanation methods

In this section, we quantitatively benchmark SEDC against existing model-agnostic explanation methods. Since LIME and SHAP generate explanations in a similar format as our EdC (i.e., a set of segments), these can be compared. We also generate explanations containing the most important segments using a ranking resulting from occluding the individual segments, which relates to the work of Zeiler et al. [52]. However, we use our segmentation instead of a squared patch for reasons of comparison.

We conduct an experiment wherein SEDC, LIME, SHAP and occlusion explanations are generated for 200 images (10 random images per class of our ImageNet data). Each image undergoes an identical segmentation for the different explanation methods. For the LIME, SHAP and occlusion explanations, the same number of segments as in the corresponding SEDC explanation is taken. For each image, 10 explanations per method (40 in total) are generated, and for each method, the following information is collected over 10 explanations:

- the stability in terms of an adapted version of the Jaccard similarity [20] (calculated as segments appearing in each of the 10 explanations divided by all unique segments in the 10 explanations),
- the average computation time,
- the counterfactual nature measured as fraction of explanations leading to a class change.

Afterwards, the information on the image explanations is aggregated over the 200 images. The results are shown in Table 4.

First, the lower stability of the generated LIME and SHAP explanations for a given image classification (respectively, 61.13% and 53.01%) points to the instability issue. In contrast, for both SEDC and occlusion, the explanations for an individual image always remain the same, since the approach is deterministic.

Second, SEDC and occlusion have lower computation times than LIME and SHAP. On average, occlusion (1.85 s per image) is generally faster than SEDC (6.57 s per image), as it more or less corresponds to the first part of the SEDC algorithm, while LIME and SHAP, respectively, need 27.38 and 33.45 s. Only in extreme cases where a high number of segments must be removed to result in a class change, SEDC takes more time than LIME and SHAP. This typically involves images wherein evidence supporting the predicted class is abundantly present and scattered across the image. Although it is possible to speed up the computation time of LIME and SHAP by reducing the number of perturbed samples, this will lower the stability of the resulting explanations even further. We note that the computation time of SEDC fluctuates more compared to the other approaches. Since the number of perturbations is chosen in advance for LIME and SHAP, the time to compute explanations for different images is similar. For the occlusion approach, this is due to the fact that each individual segment is occluded once, again resulting in a similar computation time for each image. In contrast, SEDC generates additional perturbations until explanations are found. For instance, since the explanation shown in Fig. 3c consists of two segments on a total of 37, at least 73 perturbations are made and classified (37 with one perturbed segment and 36 with two perturbed segments). This number could be higher in case other combinations with two segments were evaluated before returning to another one segment-combination in the search tree. Assuming the EdC for this image would contain five segments, at least 175 perturbations would be needed to obtain them. As a result, the computation time for SEDC is generally more volatile than for LIME, SHAP and occlusion.

Third, the most important segments resulting from LIME, SHAP and occlusion only result in a class change (and thus EdC) after removal in, respectively, 29.10%, 44.75% and 35.50% of the cases. These percentages decrease considerably in case only explanations consisting of more than one segment are considered: respectively, 25.67%, 34.49% and 15.75%. Remember SEDC always leads to counterfactual explanations, as the class change is the objective of the algorithm. This implies that, although LIME, SHAP and occlusion identify segments that support the predicted class, these are not necessarily the most discriminative ones. The results for explanations with multiple segments also stress the role of dynamics in the removal of evidence, since simply taking the top ranked segment(s) does often not result in an EdC.

5 Conclusions and future research

Explaining predictions of complex image classification models is an important and challenging problem in the machine learning field. Many existing explanation methods generate feature importance rankings, which have drawbacks regarding explanation size, feature dependence and being related to one prediction class. Even though counterfactual explanations are considered promising to explain complex model decisions, the existing counterfactual methods for image classification have strong requirements regarding disclosure of training data and model internals, which are often unrealistic in practice. Therefore, we introduced SEDC and SEDC-T as model-agnostic, instance-level explanation methods for image classification. They allow for the automatic generation of visual counterfactual explanations and address the mentioned drawbacks of existing explanation methods. We experimentally test our approach on ImageNet data and illustrate with concrete examples that the resulting explanations can give insight in model decisions. Moreover, we compare our approach qualitatively and quantitatively with popular existing explanation methods. The results point at the stability, computational efficiency and effectiveness of SEDC in finding counterfactuals.

In future research, the segmentation and segment replacement methods can be further explored and tested in different application contexts. Taking this further, a large benchmarking study that implements and compares the different building blocks of counterfactual generating algorithms, could provide more insight into what leads to better explanations (or measures), and might lead to novel, improved algorithms. This type of work was already reported in literature [14, 39] but only considering tabular data. In addition, the use of explanations in model debugging for a real-life application could reveal how data quality can be improved with the proposed SEDC-T method. Finally, to go beyond the examples presented in this paper, the qualitative evaluation of explanations by end users could be helpful to assess to what extent they lead to a higher perceived explainability in practice.

Appendix A: SEDC algorithm

Algorithm 2 SEDC algorithm

```
Inputs:
$I$ % Image to classify
$C_M : I \rightarrow \{1, 2, ..., k\}$ % Trained classifier with $k$ classes and scoring function $f_{C_M}$
$S = \{s_i, i = 1, 2, ..., l\}$ % Segmentation of image with $l$ segments

Procedure:
$c = C_M(I)$ % Predicted class
$p_c = f_{C_{M,c}}(I)$ % Score for predicted class
$R = \{\}$ % List of EdCs
$C = \{\}$ % List of combinations to expand on
$P = \{\}$ % List of predicted class score reductions
for $s_i$ in $S$ do
    $c_{new} = C_M(I \backslash s_i)$ % Class after removing $s_i$ from $I$
    $p_{c,new} = f_{C_{M,c}}(I \backslash s_i)$ % Score after removing $s_i$ from $I$
    if $c_{new} \neq c$ then
        $R = R \cup \{s_i\}$
    else
        $C = C \cup \{s_i\}$
        $P = P \cup (p_c - p_{c,new})$
    end if
end for
while $R = \emptyset$ do
    $k = \text{argmax}(P)$
    $best = C_k$ % Best-first: highest reduction in predicted class score
    $best\_set$ = all expansions of $best$ with one segment
    $C = C \backslash best$ % Pruning step
    $P = P \backslash p_k$ % Pruning step
    for $C_0$ in $best\_set$ do
        $c_{new} = C_M(I \backslash C_0)$ % Class after removing $C_0$ from $I$
        $p_{c,new} = f_{C_{M,c}}(I \backslash C_0)$ % Score after removing $C_0$ from $I$
        if $c_{new} \neq c$ then
            $R = R \cup C_0$
        else
            $C = C \cup C_0$
            $P = P \cup (p_c - p_{c,new})$
        end if
    end for
end while

Output:
EdCs in $R$
```

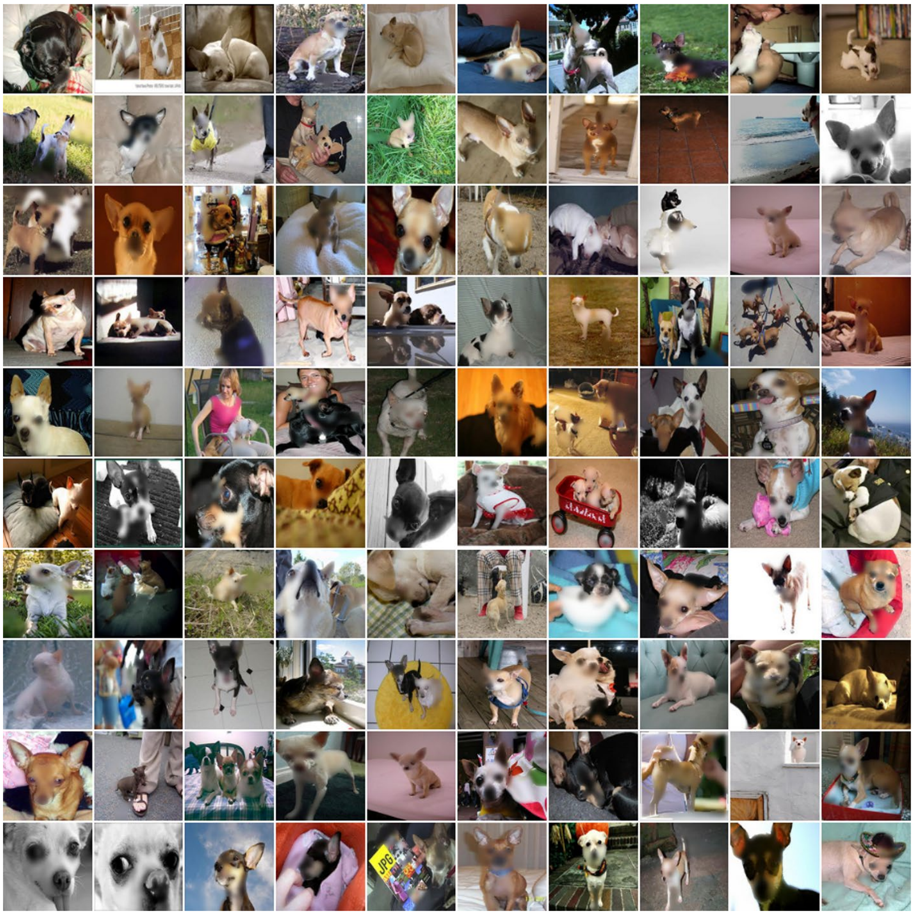


Fig. 12 Counterfactual outputs from the original class *Chihuahua* targeting the class *French bulldog* using SEDC-T

Appendix B: SEDC-T algorithm

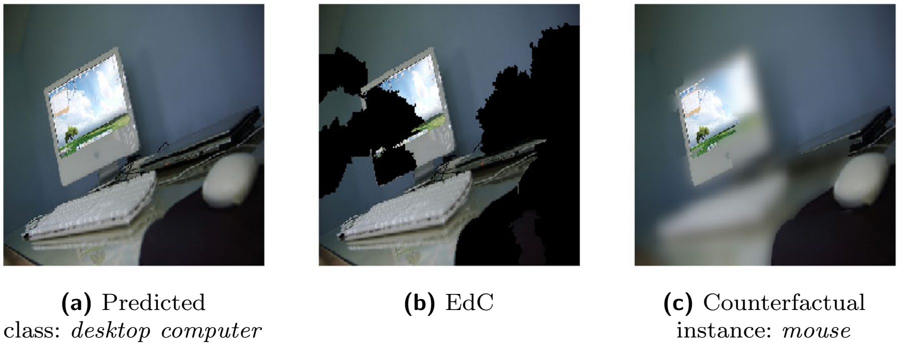
Algorithm 3 SEDC-T algorithm

```
Inputs:
$I$ % Image to classify
$C_M : I \rightarrow \{1, 2, \ldots, k\}$ % Trained classifier with scoring function $f_{C_M}$
$S = \{s_i, i = 1, 2, \ldots, l\}$ % Segmentation of the image with $l$ segments
$t$ % Target counterfactual class

Procedure:
$c = C_M(I)$ % Predicted class
$R = \{\}$ % List of EdCs
$C = \{\}$ % List of combinations to expand on
$P = \{\}$ % List of differences between target class and predicted class scores
for $s_i$ in $S$ do
    $c_{new} = C_M(I \backslash s_i)$ % Class after removing $s_i$ from $I$
    $p_{t,new} = f_{C_{M,t}}(I \backslash s_i)$ % Target class score after removing $s_i$ from $I$
    $p_{c,new} = f_{C_{M,c}}(I \backslash s_i)$ % Predicted class score after removing $s_i$ from $I$
    if $c_{new} = t$ then
        $R = R \cup \{s_i\}$
    else
        $C = C \cup \{s_i\}$
        $P = P \cup (p_{t,new} - p_{c,new})$
    end if
end for
while $R = \emptyset$ do
    $k = \text{argmax}(P)$
    $best = C_k$ % Best-first: highest difference between target and predicted class score
    $best\_set$ = all expansions of $best$ with one segment
    $C = C \backslash best$ % Pruning step
    $P = P \backslash p_k$ % Pruning step
    for $C_0$ in $best\_set$ do
        $c_{new} = C_M(I \backslash C_0)$ % Class after removing $C_0$ from $I$
        $p_{t,new} = f_{C_{M,t}}(I \backslash C_0)$ % Target class score after removing $C_0$ from $I$
        $p_{c,new} = f_{C_{M,c}}(I \backslash C_0)$ % Predicted class score after removing $C_0$ from $I$
        if $c_{new} = t$ then
            $R = R \cup C_0$
        else
            $C = C \cup C_0$
            $P = P \cup (p_{t,new} - p_{c,new})$
        end if
    end for
end while

Output:
EdCs in $R$
```

Fig. 13 Misclassification for which SEDC-T without time limit succeeds


(a) Predicted class: *desktop computer*

(b) EdC

(c) Counterfactual instance: *mouse*

Appendix C: Irreducibility of EdCs

As mentioned in Sect. 3.1, SEDC cannot guarantee that the obtained EdC is irreducible when a nonlinear model is used. This can be verified by evaluating whether any subset of the EdC leads to a class change. We applied this to 1,000 images of our data set (50 random images per label). A smaller EdC was found 195 cases, while in the other 805 cases the EdC was found to be irreducible. On average, the reduced EdCs are 39.05% smaller than the original ones. Only in 26 cases the relative reduction in size is larger than 50%. The local search lasts longer for EdCs with more segments, since more subsets must be considered. For an EdC with 13 segments, the local search takes almost 8 min on our device, since 8177 subsets need to be considered. Since the irreducibility is in most cases not violated (only 19.5% in our experiment), the relative size reduction is limited, and the additional local search can be time consuming for larger explanations, it can be argued that this step could be omitted or considered an optional post-processing step.

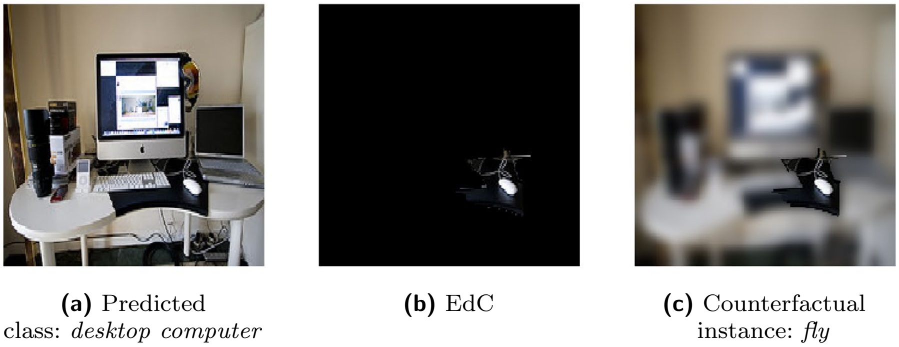
Appendix D: SEDC-T Experiment—blurred images

See Fig. 12.

Appendix E: SEDC-T for model insight in missclassifications—image with multiple objects

After a manual inspection of the data, we see that the labeling of the images seems to play a role, since the supposedly wrong label could also be considered to be correct. Images of *mouses* classified as *desktop computer* or *desk* for which SEDC-T fails, are always ones wherein both objects are present. Since in these cases the *mouse* is only a small element in the image, a lot of evidence must be removed to change the *desktop computer* or *desk* class. Likewise, a *bearskin* is typically part of a *military uniform*. In some cases, SEDC-T does find a solution when a longer search time is allowed. Remember again that this has been conducted on a basic laptop. The speed of calculations can easily be improved by performing the calculations in parallel or on a more powerful computer. Consider for instance Fig. 13, where the target class is reached after 38 s. In other cases, the target class is never reached, even without a time limit. An example is shown in Fig. 14 where, even after removing all segments except for those containing the actual mouse, the image is still not classified as a *mouse*. Although no real counterfactual explanation is found then, the evidence removed within the time limit can always be used as a partial explanation. As long as the score difference between the target and the predicted class improves, useful evidence is added to the explanation.

Fig. 14 Misclassification for which SEDC-T without time limit fails


(a) Predicted class: *desktop computer*

(b) EdC

(c) Counterfactual instance: *fly*

Table 5 SEDC-T applied to misclassifications

Label	# images	Top 5 misclassifications	# misclassified	# target found
Acoustic guitar	2015	Electric guitar	111	94
		Banjo	41	36
		Violin	21	18
		Cello	15	13
		Stage	9	6
Barrow	1334	Plow	20	16
		Park bench	17	13
		Tricycle	14	12
		Horse cart	14	13
		Stretcher	12	10
Beach wagon	1360	Minivan	99	97
		Pickup	40	39
		jeep	38	34
		Convertible	36	34
		Limousine	16	13
Beacon	1806	Breakwater	38	37
		Promontory	13	13
		Church	12	12
		Castle	11	11
		Bell cote	8	8
Chihuahua	1749	Miniature	78	72
		Toy terrier	34	33
		Italian greyhound	30	22
		Boston bull	25	21
		Pomeranian	24	18
Church	1327	Castle	83	70
		Monastery	63	58
		Altar	49	46
		Vault	44	35
		Bell cote	31	29
Envelope	1023	Wallet	28	27
		Carton	22	22
		Packet	15	15
		Handkerchief	8	8
		Binder	7	7
Espresso maker	1126	Coffeepot	43	39
		Switch	6	5
		Polaroid camera	4	3
		Drum	3	2
		Printer	3	3
Fire engine	1355	Tow truck	36	33
		Garbage truck	11	9
		Ambulance	6	5
		Thresher	6	3
		Harvester	5	5
Meerkat	2338	Mongoose	81	76
		Marmot	24	22
		Madagascar	9	7
		Megalith	7	3
		Wallaby	6	6

Table 5 (continued)

Label	# images	Top 5 misclassifications	# misclassified	# target found
Military uniform	1430	Bearskin	33	20
		Assault rifle	19	19
		Rifle	16	16
		Pickelhaube	15	13
		Stretcher	11	10
Mouse	1303	Desktop computer	65	38
		Desk	57	21
		Computer	18	17
		Laptop	15	11
		Notebook	10	6
Pencil sharpener	1268	Switch	12	12
		Rubber eraser	10	10
		Pencil box	9	9
		Polaroid camera	6	6
		Iron	6	5
Polaroid camera	1239	Reflex camera	26	24
		Printer	7	6
		Tape player	6	5
		Pencil sharpener	5	5
		Switch	4	4
Revolver	1212	Rifle	24	20
		Assault rifle	18	15
		Holster	18	15
		Pencil sharpener	4	4
		Flute	4	3
Rugby ball	1507	Soccer ball	36	29
		Volleyball	12	9
		Baseball	11	5
		Football helmet	9	7
		Cowboy hat	6	5
Soccer ball	1344	Rugby ball	11	11
		Volleyball	9	7
		Ballplayer	7	4
		Unicycle	7	4
		Croquet ball	6	6
Tiger	2085	Tiger cat	30	30
		Snow leopard	14	12
		Lynx	11	8
		Cougar	8	8
		Zebra	8	8
Toaster	1357	Microwave	17	14
		Pencil sharpener	16	15
		Rotisserie	7	7
		Switch	6	5
		Printer	6	5
Warplane	1097	Aircraft carrier	33	30
		Airliner	22	22
		Space shuttle	16	16
		Wing	12	12
		Missile	7	5

Appendix F: Experiment misclassifications

See Table 5.

Funding This work was supported by the AXA Joint Research Initiative (JRI) project 'Explainable Artificial Intelligence' and by Flanders AI Impulse Program ("Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen").

Availability of data and material We made use of publicly available data sets.

Code availability https://github.com/ADMAntwerp/ImageCounterfactualExplanations

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Süsstrunk S (2012) Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
2. Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160
3. Akula AR, Wang S, Zhu SC (2020) Cocox: generating conceptual and counterfactual explanations via fault-lines. In: AAAI, pp 2594–2601 (2020)
4. Alvarez-Melis D, Jaakkola TS (2018) On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049
5. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García S, Gil-López S, Molina D, Benjamins R et al (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 58:82–115
6. 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
7. Barocas S, Selbst AD, Raghavan M (2019) The hidden assumptions behind counterfactual explanations and principal reasons. arXiv preprint arXiv:1912.04930
8. Bertalmio M, Bertozzi AL, Sapiro G (2001) Navier–Stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol 1, pp I–I. IEEE (2001)
9. Brughmans D, Martens D (2021) Nice: an algorithm for nearest instance counterfactual explanations. arXiv preprint arXiv:2104.07411
10. Byrne RM (2019) Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI, pp 6276–6282
11. Chander A, Srinivasan R, Chelian S, Wang J, Uchino K (2018) Working with beliefs: Ai transparency in the enterprise. In: IUI Workshops (2018)
12. Chen D, Fraiberger SP, Moakler R, Provost F (2017) Enhancing transparency and control when drawing data-driven inferences about individuals. Big Data 5(3):197–212
13. Cysneiros LM, Raffi M, do Prado Leite JCS (2018) Software transparency as a key requirement for self-driving cars. In: 2018 IEEE 26th International requirements engineering conference (RE). IEEE, pp 382–387
14. de Oliveira RMB, Martens D (2021) A framework and benchmarking study for counterfactual generating methods on tabular data. Appl Sci. https://doi.org/10.3390/app11167274
15. Dhurandhar A, Chen PY, Luss R, Tu CC, Ting P, Shanmugam K, Das P (2018) Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Advances in neural information processing systems, pp 592–603 (2018)
16. Doshi-Velez F, Kortz M, Budish R, Bavitz C, Gershman S, O'Brien D, Schieber S, Waldo J, Weinberger D, Wood A (2017) Accountability of AI under the law: The role of explanation. arXiv preprint arXiv:1711.01134
17. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115–118
18. Felzenszwalb PF, Huttenlocher DP (2004) Efficient graph-based image segmentation. Int J Comput Vis 59(2):167–181
19. Fernandez C, Provost F, Han X (2020) Explaining data-driven decisions made by AI systems: the counterfactual approach. arXiv preprint arXiv:2001.07417
20. Fletcher S, Islam MZ (2018) Comparing sets of patterns with the Jaccard index. Australas J Inf Syst 22
21. Goebel R, Chander A, Holzinger K, Lecue F, Akata Z, Stumpf S, Kieseberg P, Holzinger A (2018) Explainable AI: the new 42? In: International cross-domain conference for machine learning and knowledge extraction. Springer, Berlin, pp 295–303 (2018)
22. Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572
23. Goyal Y, Wu Z, Ernst J, Batra D, Parikh D, Lee S (2019) Counterfactual visual explanations. arXiv preprint arXiv:1904.07451
24. Gunning D (2017) Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA)
25. Haddad RA, Akansu AN (1991) A class of fast Gaussian binomial filters for speech and image processing. IEEE Trans Signal Process 39(3):723–727
26. Hendricks LA, Hu R, Darrell T, Akata Z (2018) Generating counterfactual explanations with natural language. arXiv preprint arXiv:1806.09809
27. ImageNet: Download (2020). http://image-net.org/download
28. Joshi S, Koyejo O, Vijitbenjaronk W, Kim B, Ghosh J (2019) Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint arXiv:1907.09615
29. Karimi AH, Barthe G, Schölkopf B, Valera I (2020) A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. arXiv preprint arXiv:2010.04050
30. Lapuschkin S (2019) Opening the machine learning black box with layer-wise relevance propagation. Ph.D. thesis. Technische Universität Berlin

31. Lapuschkin S, Wäldchen S, Binder A, Montavon G, Samek W, Müller KR (2019) Unmasking clever Hans predictors and assessing what machines really learn. Nat Commun 10(1):1096
32. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
33. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
34. Lee TB (2019) Autopilot was active when a tesla crashed into a truck, killing driver. https://arstechnica.com/cars/2019/05/feds-autopilot-was-active-during-deadly-march-tesla-crash/
35. Lipton P (1990) Contrastive explanation. R Inst Philos Suppl 27:247–266
36. Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems, pp 4765–4774
37. Martens D, Provost F (2014) Explaining data-driven document classifications. MIS Q 38(1):73–99
38. Miller T (2018) Explanation in artificial intelligence: insights from the social sciences. Artif Intell
39. Pawelczyk M, Bielawski S, Van den Heuvel J, Richter T, Kasneci G (2021) Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms
40. Ramon Y, Martens D, Provost F, Evgeniou T (2019) Counterfactual explanation algorithms for behavioral and textual data. arXiv preprint arXiv:1912.01819
41. 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. ACM, pp 1135–1144
42. Samek W, Müller KR (2019) Towards explainable artificial intelligence. In: Explainable AI: interpreting, explaining and visualizing deep learning. Springer, Berlin, pp 5–22 (2019)
43. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
44. 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: Proceedings of the IEEE international conference on computer vision, pp 618–626
45. Shang K (2018) Applying image recognition to insurance. https://www.soa.org/globalassets/assets/Files/resources/research-report/2018/applying-image-recognition.pdf
46. Simonite T (2018) When it comes to gorillas, google photos remains blind. https://www.wired.com/story/when-it-comes-to-gorillas-google-photos-remains-blind/
47. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 23(5):828–841
48. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199
49. Van Looveren A, Klaise J (2019) Interpretable counterfactual explanations guided by prototypes. arXiv preprint arXiv:1907.02584
50. Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. In: European conference on computer vision. Springer, Berlin, pp 705–718
51. Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GPDR. Harv. JL & Tech 31:841
52. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, Berlin, pp 818–833
53. Zintgraf LM, Cohen TS, Adel T, Welling M (2017) Visualizing deep neural network decisions: Prediction difference analysis arXiv preprint arXiv:1702.04595

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.