Evolutionary Counterfactual Visual Explanation

Jacqueline Höllig[1] Steffen Thoma[1] Cedric Kulbach[1]

Abstract: The increasing success of deep learning models in recent years comes with the drawback of increasing model complexity. Due to the complexity, model insights are hard to obtain. However, understanding the underlying reasoning for a proposed decision becomes crucial in critical settings. Counterfactual explanations are among the most popular methods to interpret predictions of so-called black-box machine learning models. They provide a form of explanation intuitive to human thinking by building "what-if" scenarios. Despite their popularity for interpreting tabular data, they find limited adaption in the visual domain. Current approaches to image counterfactuals rely heavily on access to model parameters, additional training data, or surrogate models. However, access to additional information might not always be feasible. We, therefore, propose an evolutionary-based method for counterfactual image generation with a custom mutation operator based on data augmentation to overcome these limitations. We show that generating image counterfactuals solemnly on an input instance and access to the prediction function is possible and performs on par with existing methods.

Keywords: Interpretability; Counterfactuals; Evolutionary Computation

1 Introduction

Deep Learning models are at the forefront of artificial development as they allow complex decision-making and can sometimes even discover complex patterns in data that other algorithms or humans can hardly find. Due to their complexity, those models are "black-boxes" with no human-understandable explanations for their predictions. With the adaption of such algorithms to critical areas like medical diagnosis, autonomous driving, or airport security, a human-interpretable explanation becomes crucial to gain trust in these algorithms. However, most machine learning systems lack ways to make decisions transparent to humans. Currently, interest in model-agnostic techniques of explainable and interpretable machine learning is growing [LL17; RSG16; RSG18; WMR17]. Most of those approaches determine how much each feature or which feature combination contributes to a particular decision (e.g., [Ca18; RSG16]). Nevertheless, those methods fail to show how a different prediction could have been achieved. According to Miller [Mi19], an essential factor for human-understandable explanations, besides selectivity (i.e., only some causes of the prediction are shown), sociability (i.e., interactiveness), and exclusion of probability, is contrastiveness. Contrastive explanations should not explain why an event Z happened but rather why an event W happened instead.

[1] FZI Forschungszentrum Informatik, Information Process Engineering, Haid-und-Neu Str. 10-14, 76131 Karlsruhe, Germany {hoellig,thoma,kulbach}@fzi.de

A specific class of algorithms that can provide contrastive explanations are counterfactuals. Counterfactuals present a perturbation to the original input that leads to a change in the prediction of an underlying machine learning model. The roots of counterfactuals lie in causal reasoning and offer answers to the question "What if?" and "Why?". They are already in daily use in scientific and ordinary language. Therefore, they provide an intuitive concept for humans to understand. Despite many efforts to apply counterfactuals to improve the interpretability of machine learning models (e.g., [Da20; Go19; La19; LK21; MTS19; Pa21; WMR17]), most approaches are restricted to specific input data types (e.g., [Da20; Go19; MST20]), or the underlying model concepts (e.g., [Da20; Go19]). Most work focuses on tabular data (e.g., [Da20; Dh18; MST20]). The small amount of work on images uses additional information like surrogate models [Li19; LK21], access to training data [Li19; LK21], or model parameters [Go19]. However, in the real world, this additional information is seldom available. In particular, in industrial, medical, or privacy-sensitive applications, the user is often not the model developer and, thus, has no access to model parameters or the expertise to evaluate those. Furthermore, training data is often not available due to privacy-related issues. Nevertheless, validating and explaining decisions is crucial for the user to understand the model's quality, trustworthiness, and decisions.

This work develops an approach to generating model-agnostic image counterfactuals in a multi-class prediction problem. Our approach, based on NSGA-II [De02], takes on an input image and the prediction function of some black-box classifier to be explained. To summarise the main contributions of this work, we show that:

1. the counterfactual optimization problem is applicable on images.
2. data augmentation mutation enables a better search space coverage compared to uniform mutation.
3. our approach achieves state of the art results on par with the approaches of Wachter et al. [WMR17] and Van Looveren & Klaise [LK21].

2 Related Work

To obtain an in-depth understanding of black-box models and their predictions, the current research focuses shifts from classic explainable AI tools (e.g., LIME [RSG16], GradCam [Se16], SHAP [LL17], or Saliency Maps [ACJ19]) that visualize why a particular decision was taken, to counterfactuals. Counterfactuals show why a different decision was taken via alternatives, thereby providing contrastiveness.

The first steps to adapt counterfactuals from their roots in causal reasoning to a tool for understanding black-box models were taken by Wachter et al. [WMR17]. They built on the fundamentals of Pearl [Pe00] to develop a basic stochastic counterfactual generation approach. They proposed the following formulation:

$$c = arg \min_{x'} \max_{\lambda} \lambda(f(x') - y')^2 + d(x, x') \tag{1}$$

The first part pushes the models' prediction $f(x')$ on the counterfactual x' to a new target class $y' \neq y$ other than the original class y. In the second part, the distance measure d keeps the counterfactual x' close to the original instance x, λ balances the contributions of the competing terms. Extending their work towards more realistic and interpretable counterfactuals, multiple authors provide mechanisms like feature extractors [Go19; Li19], constraints [Dh18; Ka20; MTS19], or prototypes [LK21]. Sharma et al. [SHG19] built the first framework for counterfactuals applicable to various black-box algorithms and data types without the need for extensive additional information. They were able to show that their approach works for multiple data types but was unable to produce human-interpretable counterfactuals on MNIST. Dandl et al. [Da20] created a general framework for tabular data by formulating a multi-objective problem for counterfactuals solved with the genetic algorithm NSGA-II.

While counterfactuals have already been widely explored for tabular data [Da20; Dh18; Dh19; LK21; MTS19; SHG19; WMR17], less work can be found on images. Some of the model-agnostic approaches for table data have been applied to images (e.g.,[SHG19; WMR17]), resulting in more adversarial samples than counterfactuals.[2] Approaches to image-specific counterfactuals focus primarily on counterfactuals for convolutional neural networks [Go19] and learning of surrogate models [Li19; LK21].

In contrast, our approach directly operates on the input image and the classifier prediction, eliminating the need for parameter access and training surrogate models.

3 Methodology and Model

Throughout, we consider a black-box machine learning classifier $f : X \rightarrow Y$ where $x \in X$ is a set of input features ($x = \{x_1, x_2, \ldots, x_n\}$) from the feature domain X and $y \in Y$ is a vector of probability distribution ($y = \{y_0, y_1, \ldots, y_{|y|}\}$ where $\sum_{i=0}^{|y|} = 1$) over the number of classes $|y|$. In this context, black-box denotes that only the model's output y is observable. The model's inner workings are unknown. The goal of counterfactual approaches is, given an input x and a classifier f, to provide an explanation via counter-examples allowing a human to understand why classifier f chose class y for data point x and not a counterfactual class y' [WMR17].

Adapted to the image domain, it results in: Given a query image x for which a classifier f predicts the class y, a counterfactual image x' identifies how x could be changed in a proximate ($\mathbf{R_1}$) [MST20], sparse ($\mathbf{R_2}$) [MST20] and plausible way ($\mathbf{R_3}$) [La19] so that the classifier maximizes the change in the predicted class ($\mathbf{R_4}$). Proximity refers to the distance between the original instance x and the counterfactual instance x', calculated as a distance.

[2] Adversarial samples are closely related to counterfactuals. However, in contrast to counterfactuals that aim for small perceptible changes to provide useful explanations, adversarial samples aim to make the changes as small and imperceptible as possible to detect flaws in the model [PBK20].

Sparsity is the number of feature changes between x and x'. A plausible adaption indicates that the resulting x' is in distribution with the data.

3.1 Objectives

Following the definition of a counterfactual and the resulting requirements ($\mathbf{R_1}$-$\mathbf{R_4}$), the optimization problem minimizes the distance ($\mathbf{R_1}$) $d(x, x')$ between the original data point x and the newly generated counterfactual data point x' to obtain a counterfactual that is close to the original (O_1). Furthermore, to ensure sparse changes ($\mathbf{R_2}$) the optimization problem uses the L_0-norm to minimize the number of pixels subjected to change (O_2), referred to as sparsity. The third optimization objective is the output distance ($\mathbf{R_4}$) which maximizes the classification probability of the counterfactual into a target class t. Equation (2) shows the optimization problem to be minimized.

$$\begin{aligned} \min \quad & O(x) := (O_1(x, x'), O_2(x, x'), O_3(x')) \\ s.t. \quad & f(x) \neq f(x') \\ & O_1(x, x') = d(x, x') \\ & O_2(x, x') = \sum_{n=1}^{N} \mathbb{1}_{|x_n - x'_n|} \\ & O_3(x') = 1 - f(x')_t \end{aligned} \tag{2}$$

As distance measure d, most approaches to counterfactuals adapt the l_1- or l_2-norm [Dh18; WMR17]. However, on images, traditional distance functions do not sufficiently account for image similarity as it disregards the spatial relationships of images [ZB09]. Therefore we compare the mean absolute error (using l_1-norm) and the root mean squared error (using l_2-norm) with different image-based similarity indices see Sect. 4.1 and appendix A[3]. $\mathbf{R_3}$ is addressed during the algorithm design in Sect. 3.2.

3.2 Algorithm

Our algorithm combines a modified version of NSGA-II with Island Populations and an adaption of the auto-tuning approach of Castelli et al. [Ca16]. Deb et al. [De02] developed NSGA-II already in 2002. However, it is still a heavily used algorithm for Multi-Objective Optimization today, as other algorithms like indicator-based methods (e.g., SMS-EMA [EBN05], IBEA [ZK04]) rely on the additional computation of the indicator, and the results

[3] https://github.com/JHoelli/Evolutionary_Counterfactual_Visual_Explanations/blob/master/Supplementary_Material.pdf.

of decomposition-based methods (e.g., MOEA/D [ZL07], NSGA-III [DJ13]) highly depend on the shape of the Pareto front [Is17].[4]

As Equation 2 indicates, the only mandatory inputs for the algorithm are a black-box classifier f and an input instance x. Our algorithm generates an island I_i with a sub-population p_i for each class $t \in Y \backslash \{f(x)\}$ that a classifier f can classify. For each island I_i the algorithm stated in Algorithm 1 runs in parallel, allowing the creation of counterfactuals in multiple boundry directions at once. In every generation g each island I_i generates new candidates λ_i by selecting, crossing, and mutating high-performing individuals from the population p_i.

Algorithm 1 Algorithm on island I_i

Input: Population Size P, Generation G, Original Image x,
Output: Non-Dominated Set p_i

$p_i \leftarrow$ initializePopulation(x)
$cxpb_i \leftarrow$ generateRandomNumber($len(p_i)$)
$mutpb_i \leftarrow$ generateRandomNumber($len(p_i)$)
$G \leftarrow$ maximal number of generations
evaluate(p_i)
$p_i \leftarrow$ selectNSGA(p_i)
for $g \in \{0, 1, \ldots, G\} \vee hypervolume(p_i) > \theta$ **do**
    $\lambda_i \leftarrow$ selTournament(p_i)
    for j in 1,. . . ,($|\lambda_i| - 1$)) **do**
        $cx \leftarrow \frac{cxpb_i[j-1]+cxpb_i[j]}{2}$
        if $random() < cx$ **then**
            $\lambda_i[j-1], \lambda_i[j] \leftarrow$ crossover($\lambda_i[j-1], \lambda_i[j]$)
            $cxpb_i[j-1], cxpb_i[j] \leftarrow$ adapt_cxpb($\lambda_i[j-1], \lambda_i[j]$)
        end if
    end for
    for j in 0,. . . ,($|\lambda_i| - 1$) **do**
        if $random() < mutpb_i[j]$ **then**
            $\lambda_i[j] \leftarrow$ mutate($\lambda_i[j]$)
            $mutpb_i[j] \leftarrow$ adapt_mutpb($\lambda_i[j]$)
        end if
    end for
    evaluate($\lambda_i + p_i$)
    $p_i \leftarrow$ selectNSGA($\lambda_i + p_i$)
end for

The initial n individuals of an island I_i are randomly initialized with the length of the flattened input image $|x|$ along with an individual crossover rate $cxpb_i$ and mutation rate $mutpb_i$. The generated individuals are evaluated on each objective stated in Equation (2). After evaluating the individual's fitness, non-dominated sorting is applied, and the crowding distance is calculated according to NSGA-II [De02]. The assigned ranks are used as the primary criterion in the tournament selection. Thereby, two individuals are compared according

[4] For full reasoning we refer to the supplementary material A[3].

to their rank. If they have the same rank, the crowding distance is used as a secondary criterion to retain the individual lying in the less crowded region to maintain the population's diversity. The selected individuals λ_i are crossed by performing a uniform crossover [SD91]. The unified crossover modifies two individuals $\lambda_i[j] \in \lambda_i$ and $\lambda_i[j-1] \in \lambda_i$ in place by swapping attributes according to the averaged crossover probability cx of the individual. Based on the fitness of the resulting offsprings $\lambda_i[j-1]$ and $\lambda_i[j]$, a new crossover probability $cxpb[j-1]$ and $cxpb[j]$is assigned to the corresponding offspring. The selected individuals $\lambda_i[j]$ are mutated with a mutation probability $mutpb_i[j]$ by a random change of attribute. Based on the performance $mutpb_i[j]$ is adapted. The algorithm stops if it meets the desired number of generations or exceeds a hypervolume [FPL06; ZT99] threshold of θ on all islands (i.e., on all islands, the generated solutions dominate a portion of θ of the objective space). The stopping criterion is applied to all islands independently as the goal is to achieve a high-quality, non-dominated set for each of them.

3.3 Custom Operators

Some of the operators used by default in evolutionary programming are unsuitable for the stated problem, as they do not account for spatial dependencies in images or enable images to be out of distribution. In this section, we depict the adapted operators of NSGA-II.

Initialization By default, NSGA-II initializes the parent population p_i randomly [De02]. However, initializing images with traditional stochastic techniques like Random Number Generators leads to a vast search space (number candidate solutions for an image: $(width \cdot height \cdot channels)! \cdot 255!$), which slows down convergence and the probability of finding a suitable solution.

To warmstart the algorithm by introducing relevant information and enable plausible results ($\mathbf{R_3}$), we lean on the concepts of superpixels. The original image x of size $H \times W \times C$, where H is the height, W the width, and C the channels, is divided into l patches of size $\frac{H}{l} \times \frac{W}{l} \times C$ by slicing. Therefore an image x contains N patches $x = [x_1, x_2, ..., x_N]$, where a patch x_i is of size $\frac{H}{l} \times \frac{W}{l} \times C$. Each individual in a population is generated by random shuffling the patch positions i.

Mutation Traditionally, individuals are mutated to produce new offsprings that are different from their parents, thereby encouraging diversity. Using the crossover operator alone leads to decreasing diversity and often results in local optima, as only the good parts of the parents survive in each generation (premature convergence).[De99]

The proposed mutation operator aims to prevent premature convergence and include new relevant information in the population by applying data augmentation [SK19]. The idea behind using data augmentation is to make sure that the changes are still plausible (R_3) by manipulating the image with basic augmentation techniques. Only basic techniques are used, as we do not use additional data or model parameters. The

data augmentation pipeline consists of functions for Random Flip (horizontally or vertically), Random Rotation (by factor 0.2, resulting in a counterclockwise rotation by 1.25), Random Contrast (by factors between 0.1 and 1.3, resulting in each pixel being adjusted by $factor \times (x - mean\ pixel\ value\ of\ channel)$), and Zoom (with height factors between -0.7 and -0.2, resulting in a zoom-in between $[20\%, 70\%]$).

Parameter Optimization According to Hassanat et al. [Ha19], parameters of evolutionary algorithms, especially the mutation and crossover rates, impact the obtainable results and convergence speed. Tuning these parameters beforehand can result in several preliminary experiments resulting in good values before the run. However, different values of parameters might be optimal at different stages of the evolutionary process. Mutation can be good in the initial generations to quickly explore the search space, while crossover is more useful once the search process is close to the optimal solution. The proposed algorithm implements a self-adaptive parameter control on the individual level, according to Castelli et al. [Ca16]. Each individual $p_i[j]$ in a population has its own crossover probability $cxpb_i[j]$ and its own mutation probability $mutpb_i[j]$. Both are initialized with random values between 0 and 1. During crossover, two selected individuals, $\lambda_i[j]$ and $\lambda_i[k]$, generate an offspring with the probability $cxpb = \frac{1}{2}(cxpb_i(\lambda_i[j]) + cxpb_i(\lambda_i[k]))$, where the resulting offspring has the crossover probability $cxpb(\lambda_i[j]) = cxpb + r$. r is a small positive number if the fitness of the generated offspring improved due to crossover and a small negative number in any other case. During mutation, an individual mutates with its mutation rate $mut_i[j]$. The resulting individual has a mutation rate of $mutpb(\lambda_i[j]) = mutpb + r$, where r is a small positive number if the fitness of the generated offspring improves due to mutation and a small negative number in any other case.

4 Evaluation

In this section, we evaluate the performance of our counterfactual approach on the two broadly research image datasets MNIST [LCB10] and Fashion MNIST [HRV17], to answer the following research questions that aim to contribute to this work:

Q1 How does the proposed image similarity measure influence the performance of our algorithm? → Sect. 4.1

Q2 How does the proposed mutation mechanism influence the performance of our algorithm? → Sect. 4.2

Q3 How does the image counterfactual approach perform compared to other state-of-the-art methods for image counterfactuals? → Sect. 4.3

Both datasets include 60.000 training images and 10.000 test images divided into 10 classes. An image is of size 28×28 pixels. Both datasets were split into an 80/20 train/test split. The train set was only used for training the classification model, while the following experiments were run on the test set.

The classification model consists of two convolutional layers for both datasets, followed by max-pooling. The output layer is flattened and fed into a two-layer feed-forward network with ReLu activation and a softmax output layer. This model is trained for 30 epochs with a batch size of 100 on the training set. For MNIST, the model achieves a test set accuracy of 0.9921; for Fashion MNIST, an accuracy of 0.831. We run all experiments on an Intel(R) Xeon(R) Platinum 8180M CPU with 2.50GHz with 1.5 TB of RAM. The code to our evaluation is made publicly available on github[5].

4.1 Q1: Distance Function

A counterfactual optimization problem usually includes minimizing the distance to the original data. However, on images, traditional distance measures like the root mean squared error or mean absolute error do not sufficiently account for image similarity as they disregard images' spatial relationships [ZB09]. To validate our choice of distance function, we compare the mean absolute error (ME) to other popular image similarity indexes: Information Based Statistic Similarity Measure ($ISSM$) [Al19], Feature-Based similarity Index ($FSIM$) [ZGD11], Root Mean Squared Error ($RMSE$), and the Structural Similarity Index ($SSIM$) [SAU19]. All functions were inversed and mapped to the range $[0, 1]$. Appendix B[3] defines the distance measures and transformations.

For each dataset, we randomly sample 15 instances. We run the algorithm without a target direction t on every distance $d \in \{SSIM, ISSM, FSIM, RMSE, ME\}$ for the selected images and set the number of epochs to 100, as we do not want the stopping criteria to interfere. The population size was set to 1000. The evaluation criterion is the hypervolume (i.e. the search space coverage). The goal is to cover a high fraction of the search space in a small number of generations.

Fig. 1 shows the development of the hypervolume averaged over all samples from both datasets. Overall, ME has the highest search space coverage, indicating the highest likelihood of achieving good results. After 100 epochs, the hypervolume of the algorithm optimizing ME as distance reaches an average of 0.7023, the highest result for any tested distance. Further, the superiority of ME over $RMSE$ confirms Wachter et al. [WMR17]. The sparsity introducing property of the l_1-norm used in ME as distance measure is desirable for human-understandable counterfactuals, as only a small number of variables are changed. For image examples, we refer the reader to section C in the appendix[3].

[5] https://github.com/JHoelli/Evolutionary_Counterfactual_Visual_Explanations

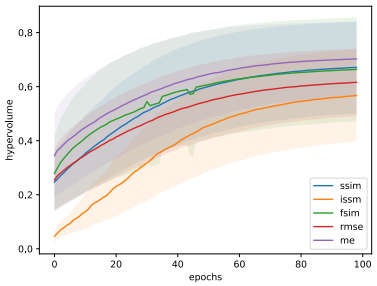


Fig. 1: Q1: Averaged hypervolume and standard deviation of experiment 1.

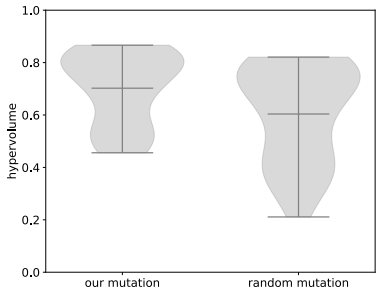


Fig. 2: Q2: Distribution of the hypervolume after 100 generations.

4.2 Q2: Mutation Operator

This section evaluates the mutation operator described in Sect. 3.3. As a baseline, we use an implementation of random mutation, replacing a pixel with a random number $r \in [0, 255]$ with a probability of 0.1.

For both mutation types, the algorithm runs on 15 randomly chosen images per dataset. With ME as distance, we run the algorithm for 100 epochs with a population size of 1000 and no target direction. Again we evaluate the hypervolume to evaluate which mutation leads better through the search space.

Fig. 2 shows the distribution of the hypervolume for our mutation and the random mutation. On average, our mutation leads to better search space coverage. It covers an, on average, over 10 % larger fraction of the search space than the random mutation baseline while having minor performance fluctuations. For image examples, we refer the reader to section C in the appendix[3].

4.3 Q3: Benchmarking

This section compares our approach to two widely used counterfactual benchmarks: the approach of Wachter et al. [WMR17] and Van Looveren & Klaise [LK21]. The approach of Wachter et al.[WMR17] is a simple stochastic optimization between the distance of the original image and the counterfactual image. Like our approach only the input image and the classification are necessary as inputs. A more sophisticated approach regarding the data distribution was developed by Van Looveren & Klaise [LK21] by training a surrogate model for counterfactual search. Therefore, Van Looveren & Klaise [LK21] approach is a slightly harder benchmark for our algorithm to meet as we do not use additional information regarding the data distribution.

For both datasets, a representative of each class is chosen, resulting in 10 images per dataset. Our algorithm runs on each image in every possible target direction $t \notin \{f(x)\}$ for 500 epochs with a population size of 1000. We ran the benchmarks in two settings:

1. without a specific target class t, to get the overall best counterfactual image.
2. with every possible target direction $t \notin \{f(x)\}$ to calculate the benchmark metrics.

The metrics were adapted and fitted to this context from [Pa21].

- Distances: We measure the distance between a counterfactual x' and the original image x with the l_0- and the l_1- norm. The l_0 norm calculates the number of pixels changed between original and counterfactual instance and is identical to the sparsity from the optimization problem ($\mathbf{R_2}$). The l_1 norm calculates the average change and is consistent with ME ($\mathbf{R_1}$).
- Redundancy: Redundancy measures the unnecessary proposed feature changes in a counterfactual, by successively flipping one value of x' after another back to x with the goal of flipping the label back from $f(x')$ to the original predicted outcome $f(x)$. If the predicted outcome does not change, we increase the redundancy counter.
- yNN: yNN (Equation (3)) evaluates the data support ($\mathbf{R_3}$) of a counterfactual based on instances from the trainings set. Ideally, a counterfactual should be close to a factual image from the same target class t. yNN is calculated by measuring how different neighborhood points around the counterfactual x' are classified. knn are the k-nearest neighbors of the original image x. We use a value of $k = 5$.

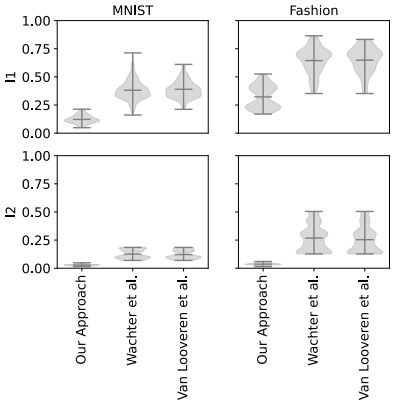


Fig. 3: Evaluation of the l_1- and l_2-distance distribution of counterfactual explanations.

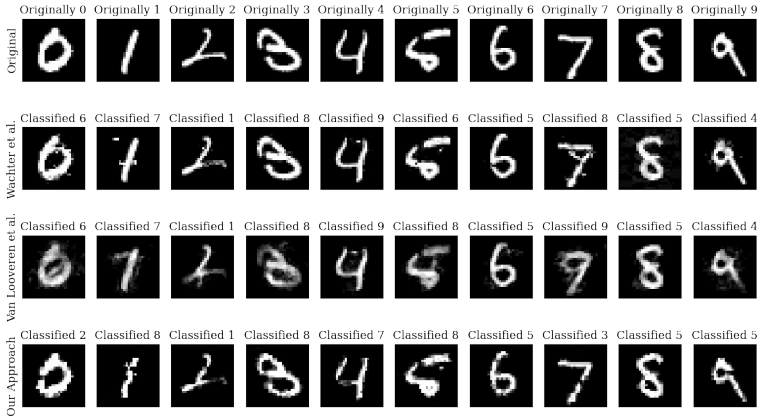
$$yNN = 1 - \frac{1}{k} \sum_{j \in kNN(x')} \mathbb{1}_{f(x')=f(x_j)} \quad (3)$$

	Method	yNN	Redundancy
MNIST	Our Approach	**0.61 ± 0.24**	**80.17 ± 41.64**
	Wachter et al.	0.48 ± 0.26	150.59 ± 54.17
	Van Looveren & Klaise	0.49 ± 0.25	158.06 ± 43.72
Fashion	Our Approach	**0.67 ± 0.24**	202.3 ± 124.31
	Wachter et al.	0.49 ± 0.26	**161.37 ± 74.89**
	Van Looveren & Klaise	0.5 ± 0.25	168.80 ± 79.66

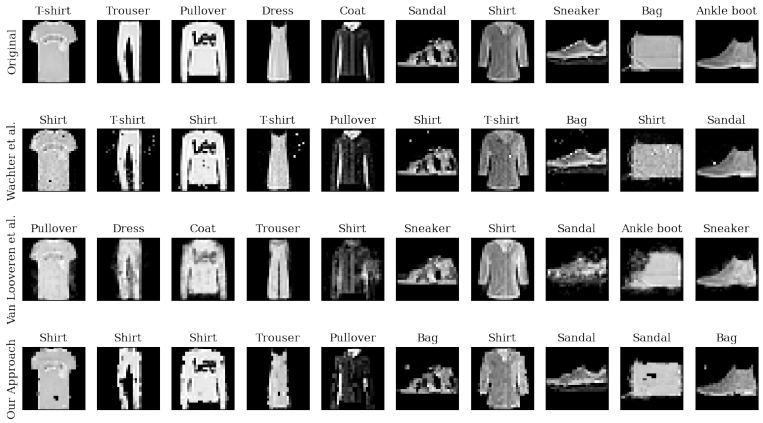
Tab. 1: Results on metrics yNN (higher is better) and redundancy (lower is better).

Fig. 3 shows that our approach achieved the on-average lowest distances, resulting in counterfactuals that have on average fewer changes (l_1) and smaller changes (l_2) than the other two approaches.

Tab. 1 adds the obtained results for the metrics redundancy and yNN. For both datasets, our approach achieves the highest yNN indicating that the resulting counterfactuals have higher support from the training data than the counterfactuals obtained by the benchmark approaches. On redundancy, no conclusion can be made. On MNIST our approach has a smaller redundancy (i.e. less unnecessary changes) than the benchmark approaches, while on Fashion MNIST the opposite is the case.



(a) MNIST



(b) Fashion MNIST

Fig. 4: Visualization of the best found counterfactual of every evaluated approach on MNIST and Fashion MNIST.

For both datasets, our approach found the closest counterfactual with the most support from the training dataset, while using no other input information than the classifiers prediction function and the input image. The redundancy is in this case negligible, as sometimes more pixels need to change to obtain higher support from training data than for changing a classifier's decision. Fig. 4 visualizes the best found counterfactual for the input images with all the evaluated approaches.

5 Conclusion

In this work, we introduced an approach to generate image counterfactuals in a multiclass classification problem by perturbing the original image with evolutionary computation and data augmentation. Based on NSGA-II, we presented a promising direction in building counterfactuals close to the original input with high data support, without the need to access additional information or model parameters. Further, we show that the counterfactual optimization problem is applicable in high-dimensional feature spaces such as images and that the mutation and augmentation of the image data enables a better search space coverage. Finally, our approach achieves state-of-the-art results on par with the approaches of Wachter et al. [WMR17] and Van Loveren & Klaise [LK21]. Based on the provided approach and the general applicability, we aim to optimize the runtime of the underlying algorithm further, investigate the mutation step and apply our method to real-world applications.

Acknowledgement

This work was carried out with the support of the German Federal Ministry of Education and Research (BMBF) within the project "MetaLearn"(Grant 02P20A013).

References

[ACJ19] Atrey, A.; Clary, K.; Jensen, D.: Exploratory Not Explanatory: Counterfactual Analysis of Saliency Maps for Deep Reinforcement Learning. arXiv Prepr. arXiv1912.05743/, pp. 1–23, Dec. 2019, arXiv: 1912.05743.

[Al19] Aljanabi, M. A.; Hussain, Z. M.; Shnain, N. A. A.; Lu, S. F.: Design of a hybrid measure for image similarity: a statistical, algebraic, and information-theoretic approach. Eur. J. Remote Sens. 52/sup4, pp. 2–15, 2019, ISSN: 22797254.

[Ca16] Castelli, M.; Manzoni, L.; Vanneschi, L.; Silva, S.; Popovič, A.: Self-tuning geometric semantic Genetic Programming. Genet. Program. Evolvable Mach. 17/1, pp. 55–74, 2016, ISSN: 13892576.

[Ca18] Carter, B.; Mueller, J.; Jain, S.; Gifford, D.: What made you do this? Understanding black-box decisions with sufficient input subsets. 22nd Int. Conf. Artif. Intell. Stat./, pp. 567–576, Oct. 2018, arXiv: 1810.03805.

[Da20] Dandl, S.; Molnar, C.; Binder, M.; Bischl, B.: Multi-objective counterfactual explanations. In: International Conference on Parallel Problem Solving from Nature. Springer, pp. 448–469, 2020.

[De02] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6/2, pp. 182–197, 2002, ISSN: 1089778X.

[De99] Deb, K.: Introduction to genetic algorithms. Sadhana - Acad. Proc. Eng. Sci. 24/4, pp. 293–315, 1999, ISSN: 02562499.

[Dh18] Dhurandhar, A.; Chen, P. Y.; Luss, R.; Tu, C. C.; Ting, P.; Shanmugam, K.; Das, P.: Explanations based on the Missing: Towards contrastive explanations with pertinent negatives. Adv. Neural Inf. Process. Syst. 2018-Decem/NeurIPS, pp. 592–603, 2018, ISSN: 10495258, arXiv: 1802.07623.

[Dh19] Dhurandhar, A.; Pedapati, T.; Balakrishnan, A.; Chen, P.-Y.; Shanmugam, K.; Puri, R.: Model agnostic contrastive explanations for structured data. arXiv preprint arXiv:1906.00117/, 2019.

[DJ13] Deb, K.; Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE transactions on evolutionary computation 18/4, pp. 577–601, 2013.

[EBN05] Emmerich, M.; Beume, N.; Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. Lect. Notes Comput. Sci. 3410/May 2014, pp. 62–76, 2005, ISSN: 03029743.

[FPL06] Fonseca, C. M.; Paquete, L.; López-Ibáñez, M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: 2006 IEEE Congr. Evol. Comput. CEC 2006. IEEE, pp. 1157–1163, 2006, ISBN: 0780394879.

[Go19] Goyal, Y.; Wu, Z.; Ernst, J.; Batra, D.; Parikh, D.; Lee, S.: Counterfactual visual explanations. In: International Conference on Machine Learning. PMLR, pp. 2376–2384, 2019.

[Ha19] Hassanat, A.; Almohammadi, K.; Alkafaween, E.; Abunawas, E.; Hammouri, A.; Prasath, V. B.: Choosing mutation and crossover ratios for genetic algorithms-a review with a new dynamic approach. Inf. 10/12, 2019, ISSN: 20782489.

[HRV17] Han, X.; Rasul, K.; Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv Prepr. arXiv1708.07747/, 2017, arXiv: /arxiv.org/abs/1708.07747 [https:].

[Is17] Ishibuchi, H.; Setoguchi, Y.; Masuda, H.; Nojima, Y.: Performance of Decomposition-Based Many-Objective Algorithms Strongly Depends on Pareto Front Shapes. IEEE Trans. Evol. Comput. 21/2, pp. 169–190, 2017, ISSN: 1089778X.

[Ka20] Karimi, A.-H.; Barthe, G.; Balle, B.; Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 895–905, 2020.

[La19] Laugel, T.; Lesot, M. J.; Marsala, C.; Renard, X.; Detyniecki, M.: The dangers of post-hoc interpretability: Unjustified counterfactual explanations. IJCAI Int. Jt. Conf. Artif. Intell. 2019-Augus/, pp. 2801–2807, 2019, ISSN: 10450823, arXiv: 1907.09294.

[LCB10] LeCun, Y.; Cortes, C.; Burges, C.: MNIST handwritten digit database. ATT Labs [Online]. Available http//yann.lecun.com/exdb/mnist 2/, 2010.

[Li19] Liu, S.; Kailkhura, B.; Loveland, D.; Han, Y.: Generative counterfactual introspection for explainable deep learning. Glob. 2019 - 7th IEEE Glob. Conf. Signal Inf. Process. Proc./, 2019, arXiv: 1907.03077.

[LK21] Looveren, A. V.; Klaise, J.: Interpretable counterfactual explanations guided by prototypes. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 650–665, 2021.

[LL17] Lundberg, S. M.; Lee, S. I.: A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. 2017-Decem/Section 2, pp. 4766–4775, 2017, ISSN: 10495258, arXiv: 1705.07874.

[Mi19] Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artif. Intell. 267/, pp. 1–38, 2019, ISSN: 00043702, arXiv: 1706.07269.

[MST20] Mothilal, R. K.; Sharma, A.; Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proc. 2020 Conf. Fairness, Accountability, Transpar. ACM, New York, NY, USA, pp. 607–617, Jan. 2020, ISBN: 9781450369367, arXiv: 1905.07697.

[MTS19] Mahajan, D.; Tan, C.; Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv preprint arXiv:1912.03277/, 2019.

[Pa21] Pawelczyk, M.; Bielawski, S.; Heuvel, J. v. d.; Richter, T.; Kasneci, G.: Carla: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. arXiv preprint arXiv:2108.00783/, 2021.

[PBK20] Pawelczyk, M.; Broelemann, K.; Kasneci, G.: Learning Model-Agnostic Counterfactual Explanations for Tabular Data. In: Proc. Web Conf. 2020. c, ACM, New York, NY, USA, pp. 3126–3132, Apr. 2020, ISBN: 9781450370233.

[Pe00] Pearl, J.: Causality. Cambridge university press, 2000, ISBN: 0521773628.

[RSG16] Ribeiro, M. T.; Singh, S.; Guestrin, C.: "Why should i trust you?Ëxplaining the predictions of any classifier. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. 13-17-Augu/, pp. 1135–1144, 2016, arXiv: 1602.04938.

[RSG18] Ribeiro, M. T.; Singh, S.; Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 32, 2018.

[SAU19] Sara, U.; Akter, M.; Uddin, M. S.: Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. J. Comput. Commun. 07/03, pp. 8–18, 2019, ISSN: 2327-5219.

[SD91] Spears, W. M.; De Jong, K. A.: On the virtues of parameterized uniform crossover, tech. rep. May, Naval Research Lab Washington DC, 1991, pp. 230–236.

[Se16] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. Int. J. Comput. Vis. 128/2, pp. 336–359, Oct. 2016, ISSN: 15731405, arXiv: 1610.02391.

[SHG19] Sharma, S.; Henderson, J.; Ghosh, J.: CERTIFAI: Counterfactual Explanations for Robustness. Transparency, Interpretability, and Fairness of Artificial Intelligence models/, 2019.

[SK19] Shorten, C.; Khoshgoftaar, T. M.: A survey on Image Data Augmentation for Deep Learning. J. Big Data 6/1, 2019, ISSN: 21961115.

[WMR17] Wachter, S.; Mittelstadt, B.; Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. Harv. J. Law Technol. 31/, pp. 1–52, Nov. 2017, arXiv: 1711.00399.

[ZB09] Zhou Wang; Bovik, A.: Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. IEEE Signal Process. Mag. 26/1, pp. 98–117, Jan. 2009, ISSN: 1053-5888.

[ZGD11] Zhang, Y.; Gong, D. W.; Ding, Z. H.: Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. Expert Syst. Appl. 38/11, pp. 13933–13941, 2011, ISSN: 09574174.

[ZK04] Zitzler, E.; Künzli, S.: Indicator-based selection in multiobjective search. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 3242/i, pp. 832–842, 2004, ISSN: 16113349.

[ZL07] Zhang, Q.; Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11/6, pp. 712–731, 2007, ISSN: 1089778X.

[ZT99] Zitzler, E.; Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. 3/4, pp. 257–271, 1999, ISSN: 1089778X.