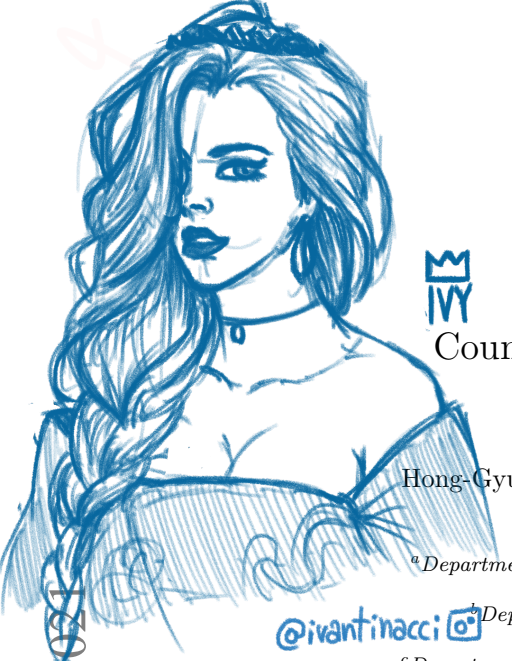


Counterfactual Explanation Based on Gradual Construction for Deep Networks

Hong-Gyu Jung[a,1], Sin-Han Kang[a,1], Hee-Dong Kim[b], Dong-Ok Won[c], Seong-Whan Lee[b,*]

[a] *Department of Brain and Cognitive Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul, 02841, Korea*

[b] *Department of Artificial Intelligence, Korea University, Anam-dong, Seongbuk-gu, Seoul, 02841, Korea*

[c] *Department of Artificial Intelligence Convergence, Hallym University, Chuncheon, Gangwon, 24252, Korea*

Abstract

To understand the black-box characteristics of deep networks, counterfactual explanation that deduces not only the important features of an input space but also how those features should be modified to classify input as a target class has gained an increasing interest. The patterns that deep networks have learned from a training dataset can be grasped by observing the feature variation among various classes. However, current approaches perform the feature modification to increase the classification probability for the target class irrespective of the internal characteristics of deep networks. This often leads to unclear explanations that deviate from real-world data distributions. To address this problem, we propose a counterfactual explanation method that exploits the statistics learned from a training dataset. Especially, we gradually construct an explanation by iterating over masking and composition steps. The masking step aims to select an important feature from the input data to be classified as a target class. Meanwhile, the composition step aims to optimize the previously selected feature by ensuring that its output score is close to the logit space of the training data that are classified as the target class. Experimental results show that our method

*Corresponding author

Email address: sw.lee@korea.ac.kr (Seong-Whan Lee)

[1] Equal contribution

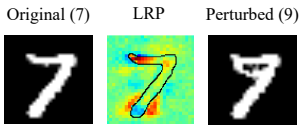
produces human-friendly interpretations on various classification datasets and verify that such interpretations can be achieved with fewer feature modification.
Keywords: explainable AI, counterfactual explanation, interpretability, model-agnostics, generative model

1. Introduction

Although deep networks exhibit remarkable performances in various tasks, the internal complexity of the models results in a transparency issue. Specifically, considering that deep networks comprise various non-linear functions, the internal mechanisms for the networks to produce an output are difficult to analyze and using such models in high risk tasks, such as credit evaluation [1, 2] and autonomous driving [3, 4] that require significant reliability and stability, is challenging owing to their lack of interpretability. In addition, the EU general data protection regulation [5] officially requires that the decision of a deep network can be explained. To comply with these technical and legal requirements, recent studies have developed algorithms that provide explanations for the decisions of deep networks.

Many of those exploit feature attribution methods [6, 7, 8, 9], which visualize the important features of an input data that lead the model to make its prediction. However, feature attribution methods are only concerned in interpreting the given input data and the predicted class; thus, the discriminative features that the model has learned to distinguish different classes cannot be directly interpreted.

To address this problem, we focus on generating counterfactual explanations. Given an input data that are classified as a class from a deep network, our goal is to perturb the subset of features in the input data such that the model is forced to predict the perturbed data as a target class. Hence, we can identify crucial features required for pre-trained networks to classify input into the target class. Fig. 1 demonstrates the difference between the feature attribution explanation and counterfactual explanation. Given the digit image with the



	Text	Prediction
Original	Perfect film from beginning to end	Positive
Perturbed	Shoddy film from beginning to end	Negative

Figure 1: (a) Comparison between counterfactual explanation and layer-wise relevant propagation (LRP) [6] that is a feature attribution method. The classification results from a pre-trained network are presented above the images. (b) Example of counterfactual explanation using the IMDB sentiment analysis dataset [10] with the prediction results.

class of "7", layer-wise relevant propagation (LRP) [6], one of the representative feature attribution methods[2], highlights the important pixels to classify it as "7" by coloring them with red. Meanwhile, our counterfactual explanation method generates a perturbed image that is classified as a target class. Thus, we can have an in-depth understanding of a network as the method verifies what a network has learned to differentiate between "7" and "9" [3]. In addition to the technical context, counterfactual explanation can be effectively used in real-world applications. For example, if a credit company using an AI system refuses a loan, our method can provide people with factors (e.g. loan amount and credit score) that are important to the assessment and which factors should be modified to meet some threshold values for loan approval. This aspect will be discussed in the experimental section.

Meanwhile, the perturbed data for counterfactual explanation should have two desirable properties. (i) **Explainability**: a generated explanation should be naturally understood by humans. Considering the example in Fig. 1(b), a pre-trained model predicts the original text as the positive class, whereas our counterfactual explanation method converts the word "Perfect" into "Shoddy" to classify the text as the negative class. Clearly, we observe that the trained model regards "Perfect" and "Shoddy" as crucial features for the prediction of positive and negative classes. (ii) **Minimality**: only a few features should be

[2]A feature attribution method is also referred to as a factual explanation method.

[3]Note that a target class is selected according to our intention to analyze the model [11].

perturbed. If we generate entirely different features from the original data to alter the original classification, the relation with the original data cannot be determined and this is, therefore, only regarded as generation but not explanation. As an example, assume that the perturbed text "*Shoddy film from beginning to end*" is changed to "*The film is shoddy*" by a counterfactual method. In this case, the changed text can produce an alternative decision but the discriminative features learned by the deep network are difficult to identify.

Although several works for counterfactual explanation have been proposed, they have limitations to employing their methods in various applications. Specifically, reference-based feature generation approaches [12, 13] were developed but the methods can be applied only to the image domain. Though domain-agnostic counterfactual explanations may be available [5, 14], they fail to satisfy the two properties and tend to provide unclear explanations that can be regarded as adversarial attack. In other words, the generated explanations appear similar to original data but are predicted as target classes. In the following sections, we verify that such a phenomenon can be resolved by considering the logit distribution of training data when generating explanation.

Herein, we propose a counterfactual explanation method based on gradual construction that considers the statistics learned from training data. We particularly generate counterfactual explanation by iterating over masking and composition steps. Given an input data, the masking step aims to select the most effective subsets of features to classify the input into a target class. To achieve this, we calculate the directive derivative with respect to the input data and choose the features that have higher sensitivity. Then, the composition step optimizes the value of the selected subsets of features by ensuring that the logit score is close to the logit distribution of the training data that are classified as the target class. This prevents the optimized features from being generated in an unpredictable distribution.

We conduct extensive experiments on text, image and finance datasets such as IMDB sentiment analysis [10], MNIST [15], HELOC [16] and UCI Credit Card [17] datasets. Experimental results show that our counterfactual method

produces human-friendly explanation using much fewer input features compared to state-of-the-art methods.

2. Related work

Many explanation methods have been developed to produce an intuitive visualization map on a given input data. Gradient-based explanation methods [6, 9, 18, 19, 20, 21] extract a representative value for each pixel by exploiting a backward operation in neural networks. Activation-based methods [22, 23, 24, 25] utilize activation maps of the convolution layer in CNNs to provide visual explanations. Reference-based explanation methods [7, 8, 26, 27, 28] compute the sensitivity of prediction scores with respect to perturbed data that are generated from masking the subset features of the original input data and replacing them with reference values such as blurred pixels, mean pixels and random noise. Furthermore, decomposition-based methods [29, 30] decompose an activation map for a classification into multiple components. Each component highlights segmented regions in the input image and has the associated importance score for the classification. In summary, all these approaches aim to generate a visualization map that highlights important regions of the input data that have impact on the prediction of deep networks. Meanwhile, we focus on creating a counterfactual explanation that indicates which features in the input data should be modified and how to generate a target class.

C. H. Chang et al. [12] recently proposed a counterfactual explanation method by masking and replacing certain image regions with artificially generated data such as blurred or generative adversarial network (GAN)-based images. Y. Goyal et al. [13] allowed a user to manually select a reference image whose prediction is a target class. Then, they aim to replace some regions of the original image with certain regions of the reference image to generate a counterfactual explanation. ABELE [31] exploits a genetic algorithm to generate several samples around an input latent vector using an auto-encoder, and employs these samples to learn a decision tree. After that, it finds a latent vec-

tor that changes from an original class to another class and decodes the vector to generate counterfactual explanation. However, these previous studies can be applied only to the image domain.

Meanwhile, there exist counterfactual explanation methods [32, 33] for a binary classifier. LORE [32] is a preceding work of ABELE [31] and generates several samples around an input without the auto-encoder to train a decision tree. By using the path of the tree, both factual and counterfactual explanations can be found. Growing sphere [33] searches for a counterfactual explanation by using the L2 distance to generate samples that gradually move away from the input and checking whether the classification result changes.

Although domain-agnostic counterfactual explanations [5, 14] have been developed to handle multi-class classification in addition to being domain agnostic, the methods tend to generate adversarial data rather than the one to interpret model characteristics as shown in Fig. 5(b). As we will present in the following sections, our method overcomes this phenomenon by generating perturbed images whose logit scores follows the logit distribution of a training dataset.

3. Methods

3.1. Problem definition

In this section, we outline a domain-agnostic method through gradual construction of counterfactual explanations. Given an input data $X \in \mathbb{R}^d$ that is classified as a class c_o under a pre-trained deep network f, we aim to perturb only the minimal subset features of X to change its decision into a target class c_t. Specifically, in order to generate a perturbed data X', we define a binary mask M and a composite $C \in \mathbb{R}^d$. The binary mask $M = \{0, 1\}^d$ indicates whether to replace subset features of X with the composite C or to preserve the features of X. The composite C represents newly generated feature values that will be replaced into a perturbed data X' instead of the original input features. Thus, we can formalize the perturbed data X' as the mask M and the composite

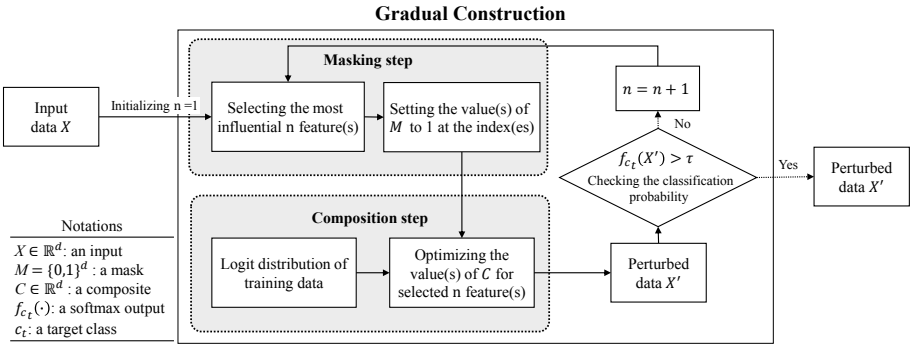


Figure 2: Overall procedure of the proposed counterfactual explanation. Our method builds gradual construction, iterating over masking steps and composite steps alternatively. These procedures are repeated until the classification probability of the perturbed data is over τ for a target class c_t.

C as follows:

$$X' = (1 - M) \circ X + M \circ C, \tag{1}$$

where M is initialized as a vector of all zeros and $\circ$ denotes the element-wise multiplication. To produce a perturbed data X' whose prediction will be a target class c_t, we progressively search for an optimal mask and a composite. To this end, our method builds gradual construction that iterates over the masking and composition steps until the desired classification score τ for the target class c_t is obtained. The goal of the masking step is to select an important feature to change the original decision into the target class c_t. After selecting the important feature, a value of mask M corresponding to the position of the feature is changed from 0 to 1. Then, the composition step optimizes a value of C for the selected feature in order to improve the output score of a target class and produce more interpretable explanations. In the following, we formally express both masking and composition steps and present our algorithm. Fig. 2 shows the overall procedure of our method.

3.2. Masking step

The goal of the masking step is to select the most influential feature to produce a target class from a pre-trained network as follows:

$$i^* = \arg\max_i f_{c_t}(X + \delta e_i), \tag{2}$$

where e_i is a one-hot vector whose value is 1 only for the i-th element, δ is a non-zero real value and f_{c_t} denotes the classification score for the target class c_t. We first suppose $\delta = \bar{\delta}h$ where h is a non-zero and infinitesimal value and $\bar{\delta}$ is a proper scalar to match the equality. Then, the objective function is approximated as the directional derivative with respect to X.

$$\begin{aligned} f_{c_t}(X + \delta e_i) &= f_{c_t}(X + \delta e_i) - f_{c_t}(X) + f_{c_t}(X) \\ &= f_{c_t}(X + \bar{\delta} h e_i) - f_{c_t}(X) + f_{c_t}(X) \\ &= \frac{f_{c_t}(X + \bar{\delta} e_i h) - f_{c_t}(X)}{h} h + f_{c_t}(X) \\ &\approx \bigtriangledown f_{c_t}(X) \bar{\delta} e_i h + f_{c_t}(X) \\ &= \bigtriangledown f_{c_t}(X) \delta e_i + R. \end{aligned} \tag{3}$$

Note that $f_{c_t}(X)$ that is not relevant to i is regarded as a constant R. Since the δ is a real value, we separately consider positive and negative cases in order to find an optimal i^*.

$$i^* = \begin{cases} max(\bigtriangledown f_{c_t}(X))_i, & if \ \ \delta > 0 \\ min(\bigtriangledown f_{c_t}(X))_i, & otherwise. \end{cases} \tag{4}$$

Here, the $max(\cdot)_i$ function returns an index that has a maximum value in the input vector and $min(\cdot)_i$ is similarly defined.

However, given that the δ value is determined at the composition step, it is not known which function should be used between the maximum and minimum operators to select the optimal i^*. Thus, we choose a sub-optimal index as

$$\hat{i}^* = max(|\bigtriangledown f_{c_t}(X)|)_i. \tag{5}$$

Although the sub-optimal choice is possible to induce more iterations for gradual construction, experimental results show that our method can efficiently produce

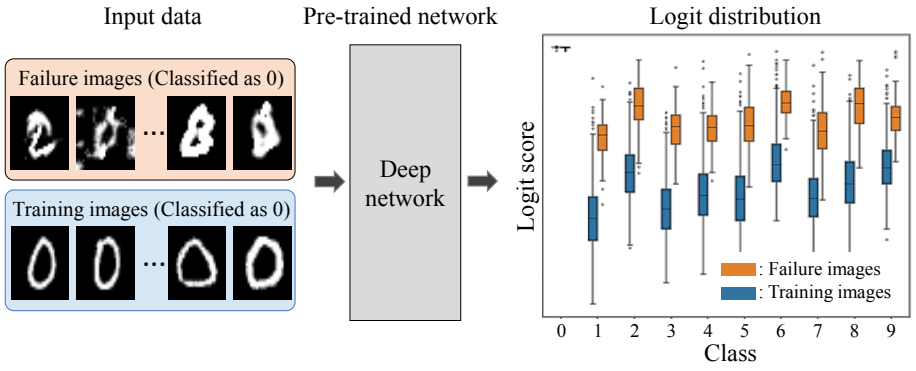


Figure 3: Comparison of logit distributions between 100 real training images and 100 failure images of CEM [14] that are classified as the class "0" under a trained model. The blue box plot is for the training data and the orange box plot represents the images of failure cases.

counterfactual explanations with fewer features than state-of-the-art methods. In summary, each masking step selects an index in the descending order by calculating Eq. 5 and changes the zero value of mask M into one.

3.3. Composition step

After selecting the input feature to be modified, the composition step optimizes the feature value to ensure that the deep network classifies the perturbed data X' as the target class c_t. To achieve this, previous works [5, 14] have proposed an objection function to improve the output score of c_t as follows:

$$\arg\max_{\epsilon} f_{c_t}(X+\epsilon) + R_\epsilon, \tag{6}$$

where $\epsilon = \{\epsilon_1, ..., \epsilon_d\}$ is a perturbation variable and R_ϵ is a regularization term.

Although it is possible for the objective function in Eq. 6 to generate interpretable counterfactual explanations, we find that it also causes an adversarial attack as shown in Fig. 5(b). To analyze the reason, we accumulated numerous failure cases of CEM [14] for the MNIST dataset. Then, we compared the distributions of logit scores (before the softmax layer) for each failure case and the training images that are classified as c_t from a pre-trained network. As shown in Fig. 3, we discovered that there exist a notable difference between the two

Algorithm 1: Gradual construction

Input:
- $X \in \mathbb{R}^d$: an input data
- c_t: a target class
- τ: the desired classification probability for the target class
- σ: the number of iteration

Initialization:
- Mask $M \in \mathbb{R}^d$ and $M_i = 0 \ \forall i$
- Composite $C \in \mathbb{R}^d$ and $C_j \sim N(0,1) \ \forall j$
- Perturbed data $X' = (1 - M) \circ X + M \circ C$
- The number of perturbed features $n = 1$

While $f_{c_t}(X') < \tau$**:**

1) Masking step

$i^* \leftarrow$ an index of the n highest value in $|\bigtriangledown f_{c_t}(X)|$

$M_{i^*} \leftarrow 1$

2) Composition step

for $m = 1$ **to** σ **do**

$$C \leftarrow \arg\min_C \left\| \sum_{k=1}^{K} \left(f'_k(X') - \frac{1}{N} \sum_{i=1}^{N} f'_k(X_{i,c_t}) \right) \right\|_2 + \lambda \left\| X' - X \right\|_2$$

$n \leftarrow n + 1$

Output: X'

distributions. When we examine the logit score distributions of the training images and failure images that are classified as a target class 0, we can observe that the logit value of failure cases from classes 1 to 9 are generally higher than the training data. Thus, we regard failure cases as the result of an inappropriate objective function that maps the perturbed data onto a different logit space from the training data. To solve this problem, we instead force the logit space of X' to belong to the space of training data as follows:

$$\arg\min_C \left\| \sum_{k=1}^{K} \left(f'_k(X') - \frac{1}{N} \sum_{i=1}^{N} f'_k(X_{i,c_t}) \right) \right\|_2 + \lambda \left\| X' - X \right\|_2 , \quad (7)$$

No. of iteration to modify

promotes to have X' to be more closer to input X

Table 1: The statistics of the IMDB, MNIST, HELOC and UCI Credit Card datasets.

Dataset	Type	# of training data	# of test data	# of classes
IMDB [10]	Text	25,000	25,000	2 (Positive/Negative)
MNIST [15]	Image	60,000	10,000	10
HELOC [16]	Tabular	7,402	2,468	2 (Approval/Refual)
UCI Credit Card [17]	Tabular	22,500	7,500	2 (Approval/Refual)

where $X' = (1 - M) \circ X + M \circ C$, K is the number of classes, f'_k represents a logit score for a class k, X_{i,c_t} denotes i-th training data that is classified into a target class c_t. N denotes the number of randomly sampled training data. In addition, to prevent from generating a totally different data from an input, we add a regularizer λ to encourage the values of X' to be close to the input data X. As a result, Eq. 7 makes the composite C to improve the probability of c_t and also pushes the perturbed data towards belonging to the logit score distribution of a training data.

Overall, gradual construction iterates over the masking and composition steps until the classification probability of a target class is reached to a hyper-parameter τ. We present a pseudo-code in Algorithm 1.

4. Experiments

We provide extensive experiments on text, image and finance datasets as follows. (1) Comparison with a feature attribution method on the text domain with the IMDB sentiment analysis dataset [10]. (2) Comparison with counterfactual explanation methods on the image domain with the MNIST dataset [15] and (3) on the finance domain with the HELOC [16] and UCI Credit Card [17] datasets. (4) Ablation study on the image and finance datasets to validate the effectiveness of our loss function that prevents a generated explanation from being adversarial data. The statistic of each dataset is provided in Table 1.

4.1. Experimental setting

We trained a multi-layer perceptron (MLP) or a convolutional neural network (CNN) that is specified below for each dataset. To generate a composite, we used the Adam optimizer [34] and set the learning rate to 0.1. Training iteration was set to $1,000$ and 500 for the MNIST dataset and the other datasets, respectively. We used the hyper-parameters N=100 and λ=0.3.

4.2. IMDB

The IMDB dataset [10] is a movie reviews dataset for sentiment classification. As we need to rely on word-to-embedding pairs for given texts, new word-to-embedding pairs for counterfactual explanation are generated to change the classification result to the alternative class.

4.2.1. Pre-trained network

Given that the dataset is composed of words, word embeddings created by GloVe [35] were used as input. Then, we trained a CNN model with three convolution layers, three max-pooling layers, a Dropout layer [36] and a fully-connected layer. The minimum word count in a movie review is restricted to five for the input. When its word number is lower than 5, the word 'pad' is added to match the requirement. In particular, this CNN model achieved a 85.4% test accuracy.

4.2.2. Algorithmic details

As we used GloVe [35], unique word-to-embedding pairs exist. However, after applying our explanation method, such embeddings are perturbed and thus, do not match the GloVe word-embedding pairs. Thus, we calculated the distance between the perturbed embedding and the GloVe embeddings. Finally, the word with the minimum distance was produced from the unique pair for explanation. As a hyper-parameter in Algorithm 1, the target probability τ is set to 0.9.

Method	Text data	Prediction
LRP	It was one of the best theatre experiences I have ever had	Positive
Ours	It was one of the dreadful theatre experiences I have ever had	Negative
LRP	This film is great and great	Positive
Ours	This film is terrible and great	Negative
LRP	The film is awful	Negative
Ours	The film is truly	Positive

Table 2: Comparison on the explanations of factual and counterfactual methods using the IMDB dataset [10]. LRP [6] that is a factual explanation method highlights important words to classify the texts. The color opacity in LRP represents the importance of the words and red and blue colors indicate positive and negative degrees. On the contrary, our method generates a counterfactual explanation by choosing the most important word and changing the word to the another to produce the alternative prediction. The changed word in our results is represented as a blue or red color.

4.2.3. Results

To compare our method with LRP [6] that is one of the representative feature attribution methods, we generated several text data for test as shown in Table 2. LRP highlights the important words of the original text which leads the model to make its prediction. To be specific, LRP in the first column of Table 2 explains that the subset words "the best theatre experiences I" in the original text contribute to produce the positive class and the most important word among them is 'best'. However, it is not clear how many words a pre-trained network needs to keep its prediction and it is not possible to explicitly know the meaning of the color opacity among the highlighted words. In contrast, our method changes 'best' into 'dreadful'. This result implies that the word 'best' is critical to classify the text to positive regardless of the other words, and the changed word 'dreadful' mainly contributes to produce an alternative class. Thus, we argue that our method can provide not only the critical regions of an input data in order to output its prediction but also what feature(s) should be changed to produce an alternative decision in terms of interpretability.

The another interesting point is illustrated in the second column of Table 2.

	Text data	Prediction
Original text	The ultimate homage to a great film actress. The film is a masterpiece of poetry on the screen. Like great poetry it is **timeless**. Direction, cast, screenplay, music, lyrics, in fact all the norms for movie...	Positive
Ours (100-dim)	The ultimate homage to a great film actress. The film is a masterpiece of poetry on the screen. Like great poetry it is **nauseating**. Direction, cast, screenplay, music, lyrics, in fact all the norms for movie...	Negative
Ours (300-dim)	The ultimate homage to a great film actress. The film is a masterpiece of poetry on the screen. Like great poetry it is **trite**. Direction, cast, screenplay, music, lyrics, in fact all the norms for movie...	Negative
Original text	I was pretty disappointed in what I believe was one of Audrey Hepburn's last movies. I 'll always love John Ritter best in slapstick. He was just too **pathetic** here...	Negative
Ours (100-dim)	I was pretty disappointed in what I believe was one of Audrey Hepburn's last movies. I 'll always love John Ritter best in slapstick. He was just too **delightful** here...	Positive
Ours (300-dim)	I was pretty disappointed in what I believe was one of Audrey Hepburn's last movies. I 'll always love John Ritter best in slapstick. He was just too **wonderful** here...	Positive
Original text	The mood of the film is captured perfectly by the camera-work and (lack of) lighting. A **great** discourse...	Positive
Ours (100-dim)	The mood of the film is captured perfectly by the camera-work and (lack of) lighting. A **dreadful** discourse...	Negative
Ours (300-dim)	The mood of the film is captured perfectly by the camera-work and (lack of) lighting. A **bad** discourse...	Negative
Original text	Yes, The Southern Star features a pretty **forgettable** title tune sung by that heavy set crooner Matt Monro. It pretty much establishes the tone for this bloated and rather dull feature, ...	Negative
Ours (100-dim)	Yes, The Southern Star features a pretty **memorable** title tune sung by that heavy set crooner Matt Monro . It pretty much establishes the tone for this bloated and rather dull feature, ...	Positive
Ours (300-dim)	Yes, The Southern Star features a pretty **memorable** title tune sung by that heavy set crooner Matt Monro . It pretty much establishes the tone for this bloated and rather dull feature, ...	Positive

Table 3: Counterfactual explanations on the texts of the IMDB test dataset [10]. The changed word in our results is represented as a blue or red color.

Our method reveals that the pre-trained model thinks only the first word 'great' as an important element to its prediction, so that substituting 'great' into 'terrible' forces the model to classify the text as the negative class. Meanwhile, the last column of Table 2 shows the weakness of our method in the text data. The perturbed word 'truly' can be considered positive by humans, but this word is grammatically incorrect. In other words, our method does not consider a grammatical error when generating counterfactual explanations.

The more qualitative results on the texts provided by the IMDB dataset are presented in Table 3. The original data is randomly selected from the test set and the highlighted part indicates the word before and after perturbation using our method. As an ablation study, we also provide the results with two different lengths of GloVe [35] embeddings. The average numbers of changed words when

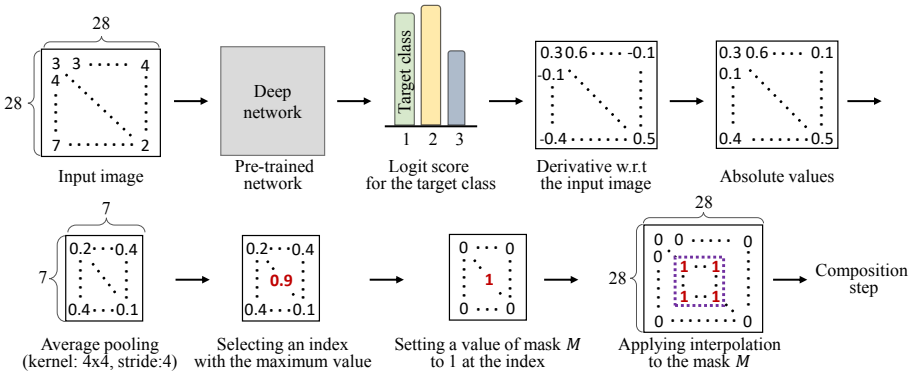


Figure 4: Detailed procedure of the masking step in the MNIST dataset [15].

using each embedding in the test set were 1.94±5.67 and 1.52±0.94, respectively. As indicated in the standard deviation, our method operates more stably with the 300-dimensional embedding due to its higher discriminative power.

4.3. MNIST

The MNIST dataset [15] is composed of hand-written digits from 0 to 9. Counterfactual explanation is provided by generating new pixels that change the classification result to a target class.

4.3.1. Pre-trained network

We trained a simple CNN for digit classification, which consists of two sets of convolution-convolution-pooling layers followed by three fully-connected layers. The CNN obtained a 98.4% test accuracy.

4.3.2. Algorithmic details

We performed a block-wise optimization for counterfactual explanation, considering that the adjacent pixels within an image share redundant information. In particular, instead of finding one pixel for the masking step, we set the dimension of the mask M to 4×4 and we optimize the 16 pixels in the composition step. Fig. 4 depicts the details of the process. Furthermore, to generate a visually smoothed image during the generation process, we added the total-variation

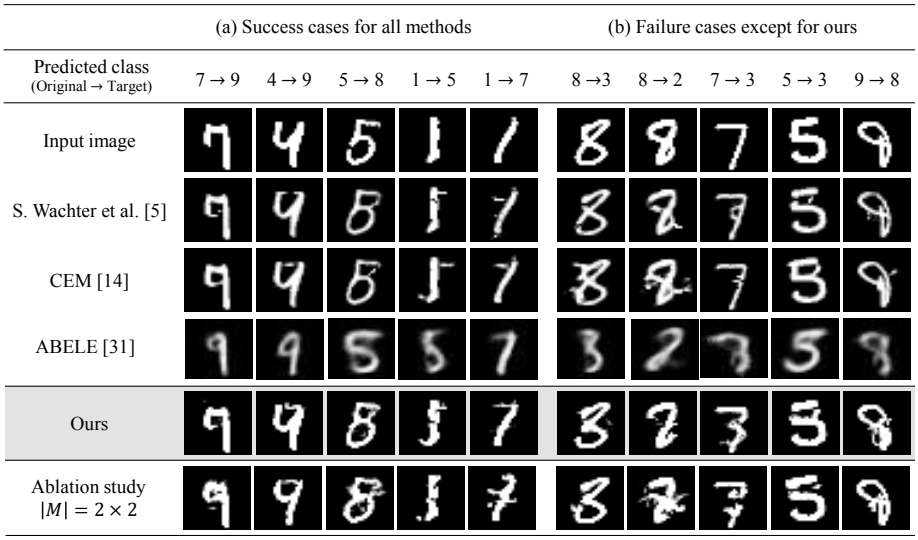


Figure 5: Counterfactual explanations on the image domain with the MNIST dataset [15]. From the input images, they generate perturbed images to classify them into a target class. (a) All methods show the success cases of counterfactual explanations. (b) Other counterfactual methods fail to provide human-friendly explanations unlike the proposed method.

regularization in Eq. 7 similar to that in [37].

$$\arg\min_{C} \left\| \sum_{k=1}^{K} f'_k(X') - \frac{1}{N} \sum_{i=1}^{N} f'_k(\bar{X}_{i,c_t}) \right\|_2 + \lambda \left\| X' - X \right\|_2 + \eta R_{tv}, \tag{8}$$

where $R_{tv} = \sum_{i,j} \left(\left(X'_{i,j+1} - X'_{i,j} \right)^{\beta} + \left(X'_{i+1,j} - X'_{i,j} \right)^{\beta} \right)^{\frac{\beta}{2}}$ and η is a hyper-parameter. The i and j are the indexes of the height and the width in an image. We set η and β to 0.3 and 2, respectively. The target probability τ is 0.9.

4.3.3. Results

For the comparison of our approach and existing counterfactual explanation methods; S. Wachter et al. [5], CEM [14] and ABELE [31], we randomly select test images and then analyze their explainability. Also, we set the target class as a similar class (e.g., $7 \rightarrow 9$, $5 \rightarrow 8$, $1 \rightarrow 7$) and a more difficult class to be changed (e.g., $1 \rightarrow 5$, $8 \rightarrow 2$, $7 \rightarrow 3$). Fig. 5(a) shows that all methods successfully

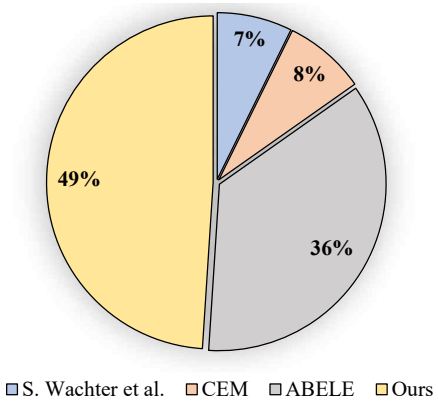


Figure 6: Evaluating how well human-friendly explanation is generated. Conducting an online survey, the statistic was obtained by calculating the percentage of counterfactual images identified as each method among the total images selected from 30 subjects. Details can be found in the text.

generate counterfactual explanations. They produce similar perturbed images for each original image and the important regions for a target class are well identified in terms of a human's point of view. However, in Fig. 5(b), S. Wachter et al. [5] and CEM [14] produce adversarial images, so that it is difficult to interpret the results and identify the discriminative regions between the two classes. Meanwhile, ABELE [31] produces blurred images. Conversely, our method exactly visualizes what features should be inserted and/or removed to be classified as the target class. It is worth noting that $|M| = 2 \times 2$ produces worse counterfactual images than $|M| = 4 \times 4$ as it increases the difficulty of the optimization by the redundancy among adjacent pixels.

In addition, to evaluate how well human-friendly explanation is generated, an online survey was configured as follows. First, 25 images were randomly selected for each algorithm to generate counterfactual images. After that, since this paper considers four algorithms, 100 counterfactual images, which were mixed randomly, were asked to check with 30 subjects whether those images looked like the same number as the ground-truth label. Finally, the statistic was obtained by calculating the percentage of counterfactual images identified

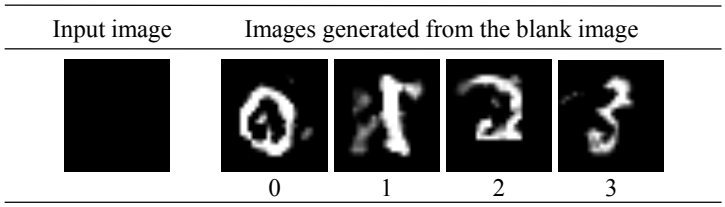


Figure 7: Generation capability of the proposed method. From the black image, our method can even generate perturbed images that seem to be target classes by using Eq. 8 except for the L2 regularization term. The target classes are presented by the numbers below the perturbed images.

as each method among the total images selected from 30 subjects. As shown in Fig. 6, the proposed method achieves the highest number as human evaluation.

To further verify the capability of feature generation, we provide experimental results by setting λ of Eq. 8 to zero in the composition step. As shown in Fig. 7, our method even converts the black image into the perturbed images that seem to be target classes. That is, rendering the logit score of a perturbed data to belong to the logit space of training images that are classified as a target class can lead to generating similar images to the training data.

4.4. HELOC and UCI Credit Card

Both HELOC (Home Equity Line of Credit) [16] and UCI Credit Card [17] are tabular datasets for loan approval/refusal classification. Normalized input features are newly generated to change the classification result to the alternative class for counterfactual explanation.

4.4.1. Pre-trained network

We normalized the input values into a range of [0, 1] using the training datasets and trained an MLP model, which is composed of five fully-connected layers. Test accuracies for the HELOC and UCI Credit Card datasets were 70.8% and 81.0%, respectively.

Table 4: Counterfactual explanations on the tabular domain with the HELOC dataset [16]. The changed feature values are represented in green.

Feature name	Input data	S. Wachter et al. [5]	CEM [14]	Ours
MSinceOldestTradeOpenv	0.427	**0.426**	0.427	0.427
MSinceMostRecentTradeOpen	0.057	**0.052**	0.057	0.057
AverageMInFile	0.465	**0.516**	**0.504**	0.465
NumSatisfactoryTrades	0.418	0.418	0.418	0.418
NumTrades60Ever2DerogPubRec	0.071	0.071	0.071	0.071
NumTrades90Ever2DerogPubRec	0	0	0	0
PercentTradesNeverDelq	0.88	0.88	0.88	0.88
MSinceMostRecentDelq	0.348	0.348	0.348	0.348
MaxDelq2PublicRecLast12M	0.667	**0.670**	0.667	0.667
MaxDelqEver	0.5	**0.498**	0.5	0.5
NumTotalTrades	0.398	0.398	0.398	0.398
NumTradesOpeninLast12M	0.125	0.125	0.125	0.125
PercentInstallTrades	0.55	0.55	**0.544**	0.55
MSinceMostRecentInqexcl7days	0.25	0.25	0.25	0.25
NumInqLast6M	0	0	0	**0.057**
NumInqLast6Mexcl7days	0	0	**-0.03**	0
NetFractionRevolvingBurden	0.507	**0.498**	0.507	0.507
NetFractionInstallBurden	0.491	**0.492**	0.491	0.491
NumRevolvingTradesWBalance	0.3	0.3	**0.298**	0.3
NumInstallTradesWBalance	0.619	**0.615**	0.619	0.619
NumBank2NatlTradesWHighUtilization	0.476	0.476	0.476	0.476
PercentTradesWBalance	0.787	0.787	0.787	0.787
Prediction	Not loanable	Loanable	Loanable	Loanable

Table 5: Counterfactual explanations on the tabular domain with the UCI Credit Card dataset [17]. The changed feature values are represented in green.

Feature name	**Input data**	**S. Wachter et al. [5]**	**CEM [14]**	**Ours**
LIMIT BAL	0.113	**0.111**	0.113	0.113
SEX	1.0	1.0	1.0	1.0
EDUCATION	0.166	**0.168**	0.166	0.166
MARRIAGE	0.333	0.333	0.333	0.333
AGE	0.148	0.148	0.148	0.148
PAY 0	0.4	**0.479**	**0.452**	**0.423**
PAY 2	0.444	0.444	**0.446**	0.444
PAY 3	0.0	0.0	0.0	0.0
PAY 4	0.0	0.0	0.0	0.0
PAY 5	0.0	0.0	0.0	0.0
PAY 6	0.0	0.0	0.0	0.0
BILL AMT1	0.026	0.026	0.026	0.026
BILL AMT2	0.042	0.042	0.042	0.042
BILL AMT3	0.076	**0.074**	0.076	0.076
BILL AMT4	0.075	**0.073**	0.075	0.075
BILL AMT5	0.132	0.132	0.132	0.132
BILL AMT6	0.397	0.397	0.397	0.397
PAY AMT1	0.0	0.0	**0.002**	0.0
PAY AMT2	0.0	0.0	**-0.012**	0.0
PAY AMT3	0.0	**0.004**	0.0	0.0
PAY AMT4	0.0	0.0	0.0	0.0
PAY AMT5	0.0	0.0	**-0.004**	0.0
PAY AMT6	0.0	0.0	0.0	0.0
Prediction	Not loanable	Loanable	Loanable	Loanable

Table 6: Quantitative evaluation using the L1, L2 and coherence metrics on the HELOC [16] and UCI Credit Card [17] datasets. The numbers indicate the mean and standard deviation.

Method	L1 metric		L2 metric		Coherence	
	HELOC	UCI Credit Card	HELOC	UCI Credit Card	HELOC	UCI Credit Card
S. Wachter et al. [5]	16.25±4.54	4.55±5.54	4.52±0.21	0.17±0.09	2.34±2.66	2.06±2.48
CEM [14]	4.12±1.89	3.12±2.36	0.27±0.17	0.20±0.11	1.27±0.36	1.94±1.96
Growing Sphere [33]	9.37±2.60	6.83 ± 2.37	0.18±0.13	0.12 ± 0.09	1.10±0.19	1.48 ± 1.17
LORE [32]	1.19 ± 0.95	1.35 ± 1.18	0.68 ± 1.04	0.37 ± 0.39	6.97 ± 3.21	6.48 ± 13.14
Ours	1.05 ± 0.10	1.01 ± 0.12	0.39 ± 0.19	0.33 ± 0.17	1.40 ± 0.17	1.29 ± 0.47

4.4.2. Algorithmic details

For these tabular datasets, Algorithm 1 is used as is, that is, without undergoing further processes for explanation. The target probability τ was set to 0.5.

4.4.3. Results

Tables 4 and 5 detail the qualitative examples and prove that our method can effectively change the decision of the pre-trained network with fewer feature modifications compared to other methods. These results can be useful when our counterfactual explanations are applied commercially in financial institutions. For example, suppose a customer who hopes to obtain a loan from a bank, but the AI system of the bank refused to grant the loan based on the record of the customer. In this situation, the customer may ask how to achieve loan approval. Fortunately, our counterfactual method can provide the important factors (e.g., loan amount and credit score) for the decision and how the values of the features should be modified for the loan approval. Furthermore, as the proposed method perturbs fewer input features than existing methods, the loan can be granted to the customer by changing only a small amount of information.

To further provide quantitative results, we introduce two L1 and L2 metrics to measure the minimality property. The L1 metric aims to count the number

of perturbed features from the original data as

$$\phi_1 = \mathbb{1}_{[0.001,\infty]}(X_{o,i} - X'_{o,i}), \tag{9}$$

where $X_{o,i}$ is the i-th element of the original data, $\mathbb{1}$ is an indicator function and the lower bound 0.001 is used to ignore noise values produced by the generative process.

The L2 metric measures the difference of a value between the perturbed data and the original data as

$$\phi_2 = ||X_{o,i} - X'_{o,i}||_2. \tag{10}$$

In addition, a robustness metric is further considered for performance comparison. Specifically, for the tabular datasets of the financial domain, it is crucial to provide two users that have similar personal information with similar counterfactual features. Thus, we added the Lipswhic estimation as

$$robustness(X_o) = \underset{X_i \in N(X_o)}{\arg\max} \frac{\|X'_i - X'_o\|_2}{\|X_i - X_o\|_2}, \tag{11}$$

where $N(X_o) = \{X_i \in \mathbb{X} \mid \sum_j \mathbb{1}_{[0.001,\infty]} (|X_{i,j} - X_{o,j}|) \leq \epsilon\}$, X_i is the instances in the test set $\mathbb{X}$ and $X_{i,j}$ is the j-th element of X_i. We call this setting coherence by following R. Guidotti *et al.* [31].

We evaluate the three metrics on $1,000$ randomly selected test samples and the experimental results are shown in Table 6. As compared to other methods, we can observe that our method generally not only uses fewer features to change the original decision on both datasets but also achieves low coherence values. In other words, our method generates similar counterfactual explanations for two analogous user information while using only a few input features.

5. Ablation study

5.1. Effect of logit distributions

To demonstrate that considering the logit distributions of the training data is crucial in generating counterfactual explanations, we compare the results using

	Text data	Prediction	Text data	Prediction
Input text	It could be one of the best movies of the year	Positive	There are many other funny scenes in this film	Positive
Ours w/o LD	It could be one of the **Feroz** movies of the year	Negative	There are many other **Equally** scenes in this film	Negative
Ours	It could be one of the **mediocre** movies of the year	Negative	There are many other **bad** scenes in this film	Negative

Table 7: Ablation study on randomly generated text data. We used a network trained on the IMDB dataset [10]. The perturbed word from an original word is represented in blue. Our method without considering the logit distribution (Ours w/o LD) produces the word that is irrelevant to the target class. However, by considering the logit distribution (Ours), the perturbed text is highly related to that of the target class.

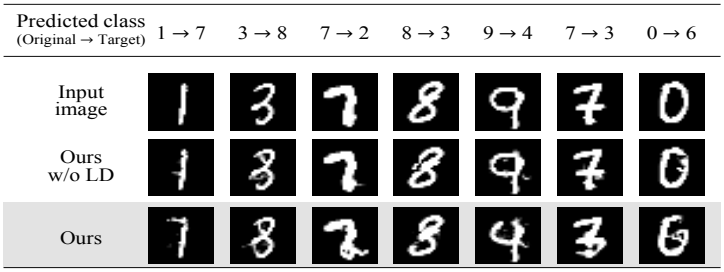


Figure 8: Ablation study on the MNIST dataset [15]. Ours w/o LD denotes the method that does not consider the logit distribution and it generates the images that are classified as the target classes but seem to be adversarial data. On the other hands, our method produces more human-acceptable images for the target classes, thus providing what features are discriminative between two classes.

our loss function in Eq. 7 with the following loss function without exploiting the logit score distribution.

$$\arg\max_{C} f_{c_t}((1-M)\circ X + M\circ C) - \lambda \left\|X' - X\right\|_2 . \tag{12}$$

Thus, Eq. 12 principally aims to increase the classification probability for c_t.

The results for the IMDB sentiment analysis dataset [10] and the MNIST dataset [15] are presented in Table 7 and Fig. 8, respectively. As shown in Table 7 (left), the method not using the logit distribution generates a perturbed text data "Feroz" that is an actor's name to be classified as the negative class.

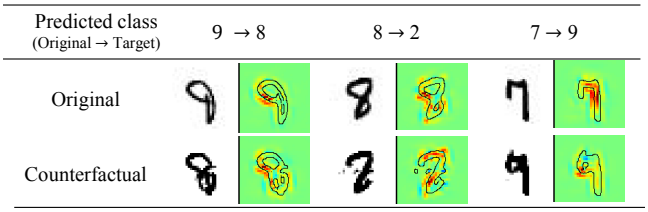


Figure 9: Applying LRP [6] to counterfactual explanation. We can observe that an factual explanation model can be combined with counterfactual images to figure out feature importance.

On the other hand, we can observe that our method changes the word in the original text into the pertinent and interpretable word to be classified as the target class. Likewise, for the MNIST dataset, the method without considering the logit distribution makes the results look like adversarial data, so that we cannot interpret which regions are discriminative between the original and target classes. Meanwhile, our method generates the results that seem to be the digit images for the target classes. To summarize, we show that the proposed loss function can prevent counterfactual explanation from being adversarial data, and generates more human-friendly interpretation for the characteristics of a pre-trained model.

5.2. Applying a factual explanation method

Counterfactual features of each class can be combined with factual explanation methods such as LRP [6] to figure out feature importance. In other words, a saliancy map for the counterfactual explanation of $8 \rightarrow 2$ should indicate important pixels to be classified as the target class 2. To this end, we show that the generated counterfactual images can be combined with external factual explanation methods in Fig. 9.

6. Discussion

As we only provide quantitative comparisons on tabular datasets, we discuss the reasons as follows. Unlike factual explanation such as LRP [6] that does

not alter input features but highlights the most important parts, counterfactual explanation should *generate* input features that change the classification result. This causes several issues for each dataset. For MNIST, as shown in the right part of Fig. 5, there exist several cases where an image, which looks like an original class but is classified as another class, is generated. In other words, we have the risk of adversarial attacks and measuring quantitative metrics such as the minimality property for non-interpretable images are not possible to provide useful information. This is clearly different characteristics from factual explanation. For IMDB, let's see a toy example. If a counterfactual method changes "the person is **awful**" to "the person is **good**" and another is to "the person is **nice**", it is unnatural to compare the distances between '**awful** – **good**' and '**awful** – **nice**'. For HELOC and UCI Credit Card, unlike the MNIST and IMDB datasets, the perturbed input features themselves contain important information. For example, if a counterfactual method suggests that the salary of a person should be changed from $2,000 to a $2,550, the difference of $550 itself is crucial for loan approval. Thus, the quality of an algorithm can be quantitatively measured by the number of changed input features and the degree of changes.

7. Limitation and future work

Although the proposed counterfactual method is model-agnostic and can handle multi-class classification, there are a few methods that can provide both factual and counterfactual explanations simultaneously, such as LORE [32] for a binary classifier and ABELE [31] for the image domain. Those methods exploit a decision tree that is itself interpretble to obtain a local decision boundary that is approximately fitted around the input. However, due to the non-linearity that makes up the feature space, *two similar inputs* can produce different local decision boundaries, resulting in *two different decision trees* and thus lower coherence values as shown in Table 6. Nonetheless, providing both factual and counterfactual explanation is highly helpful to people in various domains.

As an example, for a person who has been rejected from loan, an AI system can provide two explanations: "Your loan request has been denied as **Salary** $\leq \$1,000$ and **Age** $>$ **50** but it can be approved if **Salary** $> \$2,000$". In this view, our method only provides the latter. Thus, it is meaningful to combine the advantages of LORE [32] that can explain both factual and counterfactual factors and the proposed method that stably produces the L1 and coherence performances. We would like to leave this as future work.

8. Conclusion

In this paper, we proposed a counterfactual explanation based on gradual construction, which iterates over masking and composition steps. The masking step selects an important subset of features to classify a given input into a target class by using the directional derivative with respect to the original data. Then, the composition step updates the values of the selected features to not only improve the classification probability for a target class but also push the perturbed data towards being similar to input features. We showed that it is crucial to consider the logit distribution of training data for the composition step to prevent a perturbed data from being adversarial data. Experimental results also verified that our method satisfies explainability and minimality properties as it qualitatively provides more acceptable interpretation than the existing counterfactual explanation methods from a human's point of view and quantitatively uses fewer features for data generation.

Acknowledgment

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence, and No. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University).

References

[1] E. Rosenberg, A. Gleit, Quantitative methods in credit management: a survey, Operations Research 42 (4) (1994) 589–613.

[2] T. C. Hsu, S. T. Liou, Y. P. Wang, Y. S. Huang, et al., Enhanced recurrent neural network for combining static and dynamic features for credit card default prediction, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2019, pp. 1572–1576.

[3] H. Xu, Y. Gao, F. Yu, T. Darrell, End-to-end learning of driving models from large-scale video datasets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2174–2182.

[4] R. Garg, V. K. BG, G. Carneiro, I. Reid, Unsupervised cnn for single view depth estimation: Geometry to the rescue, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 740–756.

[5] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harvard Journal of Law & Technology 31 (2).

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, Plos One 10 (7) (2015) e0130140.

[7] R. C. Fong, A. Vedaldi, Interpretable explanations of black boxes by meaningful perturbation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3429–3437.

[8] P. Dabkowski, Y. Gal, Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, 2017, pp. 6967–6976.

[9] W.-J. Nam, S. Gur, J. Choi, L. Wolf, S.-W. Lee, Relative attributing propagation: Interpreting the comparative contributions of individual units in

deep neural networks., in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 2501–2508.

[10] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 142–150.

[11] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.

[12] C. H. Chang, E. Creager, A. Goldenberg, D. Duvenaud, Explaining image classifiers by counterfactual generation, arXiv preprint arXiv:1807.08024.

[13] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, S. Lee, Counterfactual visual explanations, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 2376–2384.

[14] A. Dhurandhar, P. Y. Chen, R. Luss, C. C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: Towards contrastive explanations with pertinent negatives, in: Advances in Neural Information Processing Systems, 2018, pp. 592–603.

[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[16] FICO, Explainable machine learning challenge, https://community.fico.com/s/explainable-machine-learning-challenge (2018).

[17] I. C. Yeh, C. H. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Expert Systems with Applications 36 (2) (2009) 2473–2480.

[18] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 3145–3153.

[19] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the International Conference on Machine Learning, 2017, pp. 3319–3328.

[20] J. Kauffmann, K.-R. Müller, G. Montavon, Towards explaining anomalies: a deep taylor decomposition of one-class models, Pattern Recognition 101 (2020) 107198.

[21] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep taylor decomposition, Pattern Recognition 65 (2017) 211–222.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929.

[23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[24] A. Chattopadhay, A. Sarkar, P. Howlader, V. N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: Proceedings fo the IEEE Winter Conference on Applications of Computer Vision, 2018, pp. 839–847.

[25] D. Liu, L. Zhang, T. Luo, L. Tao, Y. Wu, Towards interpretable and robust hand detection via pixel-wise prediction, Pattern Recognition (2020) 107202.

[26] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[27] R. Fong, M. Patrick, A. Vedaldi, Understanding deep networks via extremal perturbations and smooth masks, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2950–2958.

[28] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, S. Behnke, Interpretable and fine-grained visual explanations for convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9097–9107.

[29] B. Zhou, Y. Sun, D. Bau, A. Torralba, Interpretable basis decomposition for visual explanation, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 119–134.

[30] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J. K. Su, This looks like that: deep learning for interpretable image recognition, in: Advances in Neural Information Processing Systems, 2019, pp. 8928–8939.

[31] R. Guidotti, A. Monreale, S. Matwin, D. Pedreschi, Black box explanation by learning image exemplars in the latent feature space, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2019, pp. 189–205.

[32] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intelligent Systems 34 (6) (2019) 14–23.

[33] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Comparison-based inverse classification for interpretability in machine learning, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2018, pp. 100–111.

[34] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[35] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958.

[37] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196.