




Counterfactual explanation of Bayesian model uncertainty

Gohar Ali¹ · Feras Al-Obeidat² · Abdallah Tubaishat² · Tehseen Zia³  · Muhammad Ilyas⁴ · Alvaro Rocha⁵

Received: 11 June 2021 / Accepted: 8 September 2021 / Published online: 24 September 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Artificial intelligence systems are becoming ubiquitous in everyday life as well as in high-risk environments, such as autonomous driving, medical treatment, and medicine. The opaque nature of the deep neural network raises concerns about its adoption in high-risk environments. It is important for researchers to explain how these models reach their decisions. Most of the existing methods rely on softmax to explain model decisions. However, softmax is shown to be often misleading, particularly giving unjustified high confidence even for samples far from the training data. To overcome this shortcoming, we propose Bayesian model uncertainty for producing counterfactual explanations. In this paper, we compare the counterfactual explanation of models based on Bayesian uncertainty and softmax score. This work predictively produces minimal important features, which maximally change classifier output to explain the decision-making process of the Bayesian model. We used MNIST and Caltech Bird 2011 datasets for experiments. The results show that the Bayesian model outperforms the softmax model and produces more concise and human-understandable counterfactuals.

Keywords Deep learning · Counterfactual explanation · Bayesian model uncertainty

1 Introduction

Artificial intelligence (AI) systems are increasingly becoming ubiquitous in many domains and accepted as an effective tool for large scale automation. It is playing a vital role in a low-risk environment, such as chatbots [1] and video games [2] as well as a high-risk environment,

such as self-driving cars [3], credit lending [4], biometric recognition [5] and healthcare treatments [6]. However, the decisions made by these systems are difficult to interpret due to the opaque nature of the deep neural networks (DNN) [7]. The author in [8] discussed an example where a classifier is trained and used to differentiate between friendly and enemy tanks. It worked well on training and

✉ Tehseen Zia
tehseen.zia@comsats.edu.pk

Gohar Ali
aligohartech@gmail.com

Feras Al-Obeidat
feras.al-obeidat@zu.ac.ae

Abdallah Tubaishat
abdallah.tubaishat@zu.ac.ae

Muhammad Ilyas
muhammad.ilyas@uos.edu.pk

Alvaro Rocha
amr@iseg.ulisboa.pt

² College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates

³ National Center for Artificial Intelligence, Saudi Arabia and Department of Computer Science, Prince Muhammad Bin Fahad University, COMSATS University Islamabad, Islamabad, Pakistan

⁴ Department of Computer Science, University of Sargodha, Sargodha, Pakistan

⁵ University of Lisbon, ISEG, Rua do Quelhas No 6, Lisboa 1200-781, Portugal

¹ COMSATS University Islamabad, Islamabad, Pakistan

validation data, however, performed poorly in a real-world scenario. After investigation, it has been found that pictures of friendly tanks were taken on a sunny day, while pictures of enemy tanks were taken on a cloudy day. The classifier distinguishes tanks based on the features of the weather rather than on the features of tanks. This behaviour is particularly problematic in a high-risk environment where AI plays an important role in the decision-making process. This raises the questions: how do these systems reach their decisions, and what features do these systems consider in the input to produce a specific output? Therefore, we need explainable artificial intelligence (XAI) models [9–13] which can explain the outcomes produced by these AI systems.

Explainability in AI is generally about making an interpretable model or defining an ante-hoc or post-hoc explanation for the black-box predictor. XAI aims to make the hidden mechanism of prediction transparent and explainable to expert and non-expert users [14–16]. The counterfactual explanation is a method in XAI that produces an explanation for the decision making of DNN. In the real world, it can be defined as imagining a hypothetical scenario that is in contrast to the real observed event, such as “if X had not occurred, Y would not have occurred”. A counterfactual explanation of model prediction describes the minimal change in the input features that change the model output [17].

Existing counterfactual explanation approaches use softmax to visualize the input features or regions vital for classification. Nevertheless, it has been shown that softmax is not a true measure of model confidence, [18], its predictive probabilities are often misleading, particularly gives unjustified high confidence for samples far from the training data. In contrast, it has been shown that Bayesian uncertainty [19] provides true confidence in the model predictions. In light of these developments, we argue that if softmax provides misleading predictive probabilities, how can a counterfactual explanation relied on softmax be truthful? In this work, our objective is to produce a counterfactual explanation of Bayesian model uncertainty and analyze it in comparison with a softmax based counterfactual explanation. There are two approaches to counterfactual explanation, model-specific, also known as ante-hoc, and model-agnostic, also known as post-hoc. Model-specific deals with the internal structure of the model, while model-agnostic deals with the input and output of the model. We will follow the model-agnostic approach to produce a counterfactual explanation of the Bayesian CNN image classifier.

2 Related work

Model interpretability techniques are being used for information discovery and study. The model learns features from the dataset and becomes an information source. As a result, a model’s observations must be interpretable in order to discover inferred information for scientific findings. In this section, we’ll talk about deep learning methods for interpretability (e.g., CAM and Grad-CAM), which fall under the ante-hoc approach, and counterfactual-based model explanations, which fall under the post-hoc approach.

2.1 Ante-hoc/model-specific approach

There are multiple approaches used for explanation. They are broadly divided into two categories: a model-specific and a model-agnostic approach. A model-specific is an approach where model internal layers are accessed and modified for an explanation of a model. One technique for explaining the neural network is by visualizing their activation. The authors of [20] used the same technique and proposed Class Activation Mapping (CAM). In CAM, after the last convolution layer in the network, the flatten layer is replaced with a global average pooling (GAP) layer and linked to a dense output layer. The output layer’s class-specific weights are multiplied by the output of the last convolution to create a heat mask.

CAM, on the other hand, aids in the visualisation of features learned by the final convolution layer. Grad-CAM [21] was introduced to visualise the feature maps at each layer. To achieve this, backpropagation is used at the desired layer in the network with respect to the output class, as shown in the figure. We shall compute the gradient of class with respect to the activation maps at a specified hidden layer for this reason. The activation map is generated by multiplying these gradients with the feature maps and then using the rectified linear unit (ReLU). The output size of the chosen convolution layer determines the size of the created feature map. As a result, this map will need to be resized to fit over the original image.

Grad-CAM resizes the produced feature map and produces a low-resolution map to create the final feature map. To deal with this issue, guided backpropagation was used. The map produced by guided backpropagation and grad-CAM is merged using a point-wise operation to produce a smooth feature map. Another type of visualisation technique is backpropagating gradients back to the input image to create saliency maps. Excitation Backprop [22] and Guided Backprop [23] are some of the methods that use this technique. On various image recognition benchmarks, the authors of [23] found that max-pooling may be simply

replaced by a convolution layer with higher stride without losing accuracy. They provide a new architecture that is entirely made up of convolution layers and yields competitive results.

2.2 Post-hoc/model-agnostic approach

One of the other approaches is model-agnostic, which explains model prediction by dealing directly with the model’s input and output. The authors of [24] generate a saliency map of input (i.e., pixel/image patch) and feed it to the network as unseen or marginalize it out, calculating the difference in classification accuracy. They perform these steps on every region of the image, which increases the computation complexity. In [25], a framework for learning explanation and a paradigm for image saliency are proposed. The method learns where the algorithm focusses and discovers which features of the image most change the model accuracy when perturbed. They apply three ad-hoc in-filling methods for selecting the reference image: constant value (mean of the pixels), Gaussian noise and blur input. The authors in [26] also perform the same technique and mask the salient parts of the image to manipulate the score of the classifier. They trained an auxiliary neural network to learn that salient features mask. Their method amortizes the cost of perturbations. In contrast, the authors of [27] applied generative in-filling then optimized the network to find the image region which mostly changes the classifier output. They argue that perturbing the images makes the input far from the training data distribution, while generative in-filling produces images near the training data distribution. The authors of [28] explain the decision-making process of a deep convolution neural network (CNN) through Fault-line. The Fault-line explanation looks for the semantic level feature (e.g., horns of sheep, pointed ears of dogs) and modifies it to manipulate the classifier output.

In this work, we have followed the model-agnostic approach to produce a counterfactual explanation of the Bayesian CNN image classifier. Our objective is to produce a counterfactual explanation of Bayesian model uncertainty and analyze it in comparison with a softmax-based counterfactual explanation. The following are our contributions to this work.

- To propose a Bayesian model uncertainty based counterfactual explanation approach and related model architecture.
- To analyze the efficacy of counterfactual explanation of Bayesian model uncertainties in comparison to softmax based explanation.

3 Methodology

The authors in [26] proposed two objectives for computing the saliency map, the smallest deletion region (SDR) and the smallest supporting region (SSR). SDR answers the question: what the smallest region of input is if replaced with the reference values (in-filling techniques), the classifier score is minimized. On the other hand, SSR instead poses the question: what the smallest region of the input is if preserved or substituted into the reference input, the classifier score is maximized. In this study, we have provided results for both SSR and SDR objectives. The system architecture is illustrated in Fig. 1.

Following the approach of marginalizing and preserving a part of the image from [27], consider an input image x consisting of set of pixels Q , a Bayesian classifier gives output uncertainty $\mathcal{PB}(c|x)$ on class label c given input x and a counterfactual mask generation network $\mathcal{C}(z|x)$, shown in Fig. 2 gives mask z on given input x . The function $\mathcal{PB}(c)$ represent the Bayesian classifier confidence on class c . The subset of the input pixels are denoted by r which implies a part of the input $x = x_r \cup x_{\setminus r}$. Here, we refer to the feature map region as x_r which will be marginalized or given as unobserved to the classifier and the remaining region is referred to as $x_{\setminus r}$. Our interest is in the classifier output when x_r is treated as unobserved. The marginalization can be expressed as

$$\mathcal{PB}(c|x_{\setminus r}) = \mathbb{E}_{x_r \sim p(x_r|x_{\setminus r})}[\mathcal{PB}(c|x_{\setminus r}, x_r)] \tag{1}$$

$PG(x_r|x_{\setminus r})$ represent the in-filling function, which will approximate the x_r region to the given input $x_{\setminus r}$. The

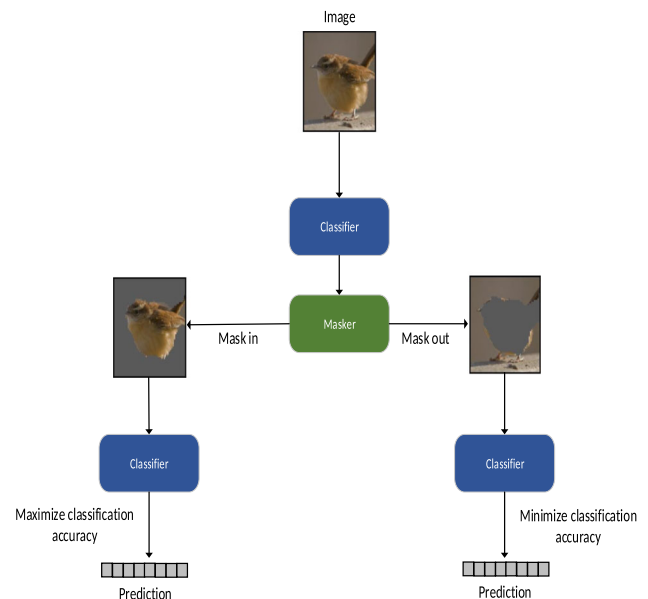


Fig. 1 System model

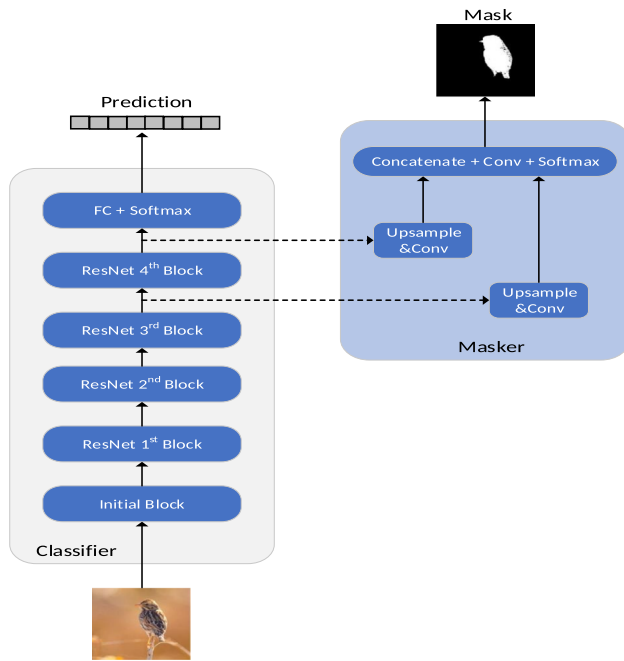


Fig. 2 Architecture of mask generation network

counterfactual generation network $\mathcal{C}(z|x)$ will output a binary mask $z \in \{0, 1\}^Q$. The infilling function φ will be the mixture of reference values and the original input image x . The function φ can be expressed as

$$\varphi(x, z) = z \odot x + (1 - z) \odot \hat{x} \text{ where } \hat{x} \sim PG(\hat{x}|x_{z=0}) \quad (2)$$

The aim is to generate mask $\mathcal{C}(z|x)$ when little numbers of reference pixels are mixed to it the output of $\mathcal{SB}(c)$ is minimized in the SDR objective. While in the SSR objective, the aim is to produce a mask $\mathcal{C}(z|x)$ when small numbers of reference pixels are mixed to it the output of $\mathcal{SB}(c)$ is maximized.

4 Results and experiments

4.1 Caltech-UCSD Birds-200-2011

For fine-grained visual categorization tasks, the CUB-200-2011 dataset (Caltech-UCSD Birds-200-2011) is one of the most often used datasets. It is the commonly preferred dataset for model interpretability as it consists of complex bird features and has resemblance between different classes, which makes it best for our counterfactual explanation where removing some features can cause the image to be misclassified by the classifier. The other reason for using this dataset is that it has a small number of samples and it belongs to bird species, which is the class in the imagenet dataset on which our basemodel is trained, thus making it the best choice for transfer learning. The dataset consists of

11,788 bird images separated into 200 subcategories, with 5,994 for training and 5,794 for testing. Each image contains 1 subcategory label, 15 part positions, 312 binary attributes, and 1 bounding box. Fine-grained natural language descriptions are added to the dataset. Each picture is given a total of ten one-sentence descriptions. Amazon's Mechanical Turk (AMT) tool is used to capture natural language details.

4.2 MNIST dataset

The MNIST dataset is one of the basic and most widely used datasets in image classification and other machine learning techniques. It has been commonly used for validating models because of its simplicity. In the experiments, we performed a contrastive explanation on the MNIST dataset, which makes it easy to understand the visualization of features learned by the different networks. The MNIST database was created using binary pictures of handwritten digits (0 through 9) from NIST's Special Database 3 and Special Database 1. The database contains a training set of 60,000 samples and a test set of 10,000 cases. It's a subset of NIST's larger database. The digits in a fixed-size image are size-normalized and centered.

4.3 Model training

First, we used the transfer learning technique to train the classifier separately on the bird dataset, and then we combined the classifier and masker to produce counterfactual instances. The classifier was developed using the Stochastic gradient descent (SGD) optimizer with a learning rate of 0.001 and a momentum of 0.9. Based on the highest test score, the best classifier is selected. The Adam optimizer is used to train the masker network, with a learning rate of 0.001. The network is trained on a GPU-based desktop machine with Nvidia TitanX Pascal (12 GB VRAM), 128 GB RAM and a 10-core Intel Xeon processor.

4.4 Transfer learning on ResNet-50

We start with a Resnet-50 model that has been pre-trained on the imagenet dataset. We performed transfer learning and trained the classifier on a bird dataset. The Dropout was also used to train the same model on a bird dataset, with the dropout at test time being used to approximate it to a Bayesian model. To compare the results, we attempted to achieve the same accuracy on both models. Table 1 shows the accuracy, macro-average, and weighted average score of Bayesian and softmax classifiers, respectively. The precision, recall, and F1 scores for each class are calculated separately, and then the macro-average and weighted

Table 1 Accuracy, macro-average and weighted average of softmax and Bayesian classifier on Birds-200-2011 dataset

	Softmax			Bayesian			Support
	Precision	Recall	F1	Precision	Recall	F1	
Accuracy			0.76			0.77	5794
Macro-average	0.77	0.76	0.76	0.78	0.77	0.77	5794
Weighted average	0.77	0.76	0.75	0.78	0.77	0.76	5794

Table 2 Bayesian vs softmax classifier accuracy scores on mask in, mask out and infilled techniques

	Softmax	Bayesian
Mask in accuracy ↑	63.46	65.60
Mask in loss ↓	1.36	1.31
Mask out accuracy ↓	7.87	8.23
Mask out loss ↑	4.38	4.35
Infilled accuracy ↓	0.82	1.29
Infilled loss ↑	5.50	5.57

average for each precision, recall, and F1 score are computed. The comparison of both classifiers shows that they perform nearly identically in terms of accuracy.

4.5 Experiments on caltech-UCSD Birds-200-2011 dataset

Table 2 shows the quantitative findings for the dataset. The Bayesian model outperforms the softmax model in the mask-in (SSR) objective, while in the mask-out (SDR) and infilled techniques, softmax performs slightly better. That

is because of the large portion of the image softmax model crop as shown in Fig. 3. Table 2 shows that the softmax model achieves 63.46% accuracy on the mask-in objective, while the Bayesian model achieves 65.60%. On the mask-out objective, the Bayesian model achieves 8.23% accuracy, which is slightly higher than the softmax model 7.87% accuracy. The softmax model on the mask-out objective crop maximum region yields lower accuracy than the Bayesian network, where the mask-out region is minimum as shown in Fig. 4 and accuracy is comparable to the softmax model. Here, for better and transparent comparison, we have shown Top 5 accuracy for both the classifiers in Table 3.

The Bayesian model performed well and produced a more accurate and plausible mask. As it has been shown by [18, 29], the softmax is not a true measure of model confidence and the softmax based models are point wise estimators, where the input is multiplied with the weights learned by the back propagation to produce the output. That output does not provide model uncertainty, which in turn generates a mask that contains more unnecessary regions. On the other hand, the Bayesian model gives uncertainty and produces a more concise mask.

Fig. 3 Softmax model generated mask and the resulted mask, mask out and infilled images are shown, the network include more surrounding pixels and generate less effective saliency map



Fig. 4 Bayesian model generated mask include less surrounding pixels and generate more effective saliency map

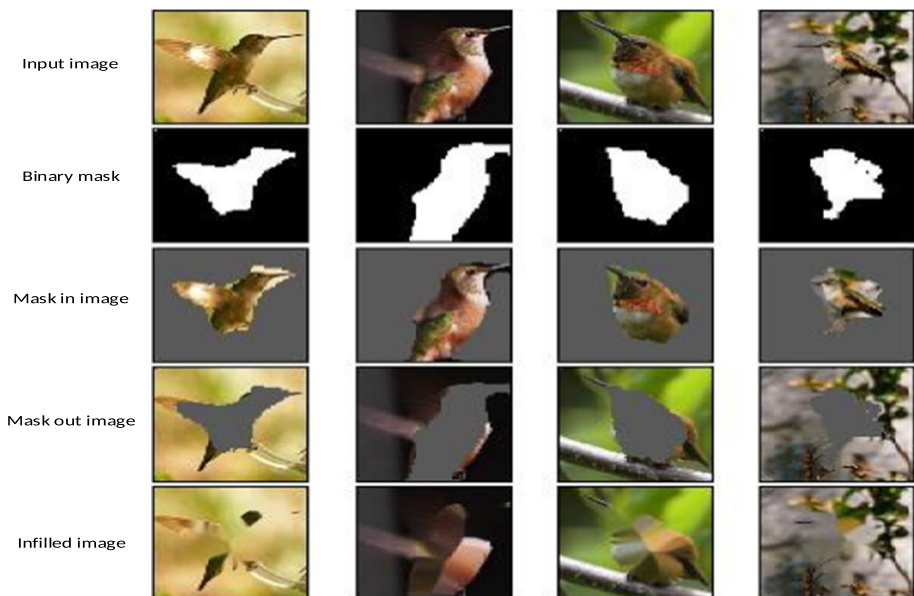


Table 3 Top 5 accuracy comparison of Bayesian and softmax classifier scores on mask in, mask out and infilled techniques

	Softmax	Bayesian
Mask in accuracy ↑	89.67	90.17
Mask out accuracy ↓	23.95	24.45
Infilled accuracy ↓	5.07	5.54

Table 4 Accuracy of softmax and Bayesian classifier on MNIST dataset

	Softmax	Bayesian
Accuracy	0.98	0.97

4.6 Experiments on MNIST dataset

This section provides quantitative and qualitative counterfactual explanation results for the mnist dataset, as well

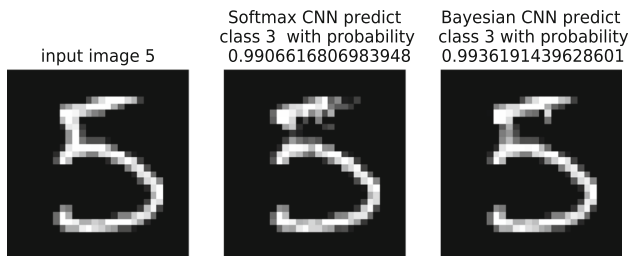


Fig. 5 Softmax vs Bayesian counterfactual instances

as a comparison of softmax and Bayesian classifiers. We created a simple CNN architecture for MNIST dataset classification and trained two instances of the network: one that predicts softmax and the other that predicts Bayesian uncertainty using the dropout approximation method. The accuracy of both networks is shown in Table 4. Because of the uncertainty, there is a little variation in the Bayesian network accuracy each time we test on a test dataset. However, we recorded 97%, which is the nearest one to the softmax model. The softmax classifier accuracy is 98% of the test dataset. Figure 5 shows the input image with softmax and Bayesian based counterfactual instance. We input the image of digit 5 to the network and the network tried to convert it to digit 3. The network applies the learned features of 3 to the image and changes it to the digit 3 with minimal change to the features. The confidence of the classifier for both converted images is shown in Fig. 5. Both networks have predicted class 3 with almost the same confidence. The difference between both generated images is shown in Fig. 8 where we can see that the feature generated by the Bayesian network Fig. 8b to change the network prediction to class 3 is more human-understandable as compared to the softmax generated features Fig. 8a. The softmax generated counterfactual instance, on the other hand, is noisy and requires more feature changes to convert the image to class 3. The process of generating counterfactual explanations for softmax and Bayesian networks is shown in Figs. 6 and 7, respectively. The initial iterations seek out counterfactuals that are outside of the distribution, whereas subsequent iterations make the counterfactuals more sparse and understandable. The findings and visualization show that the Bayesian model

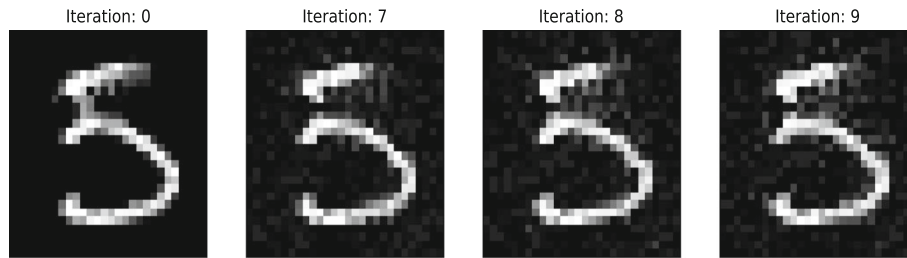


Fig. 6 Process of generating counterfactual of softmax classifier

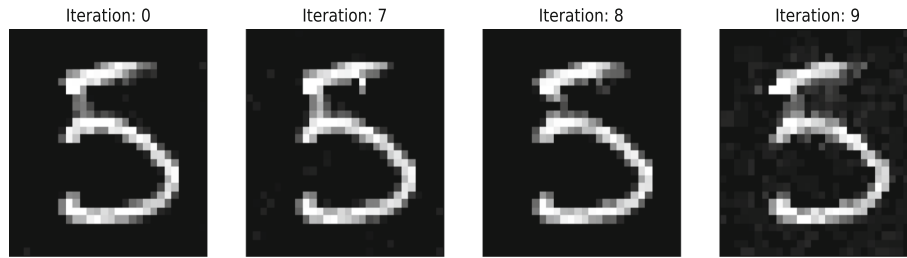
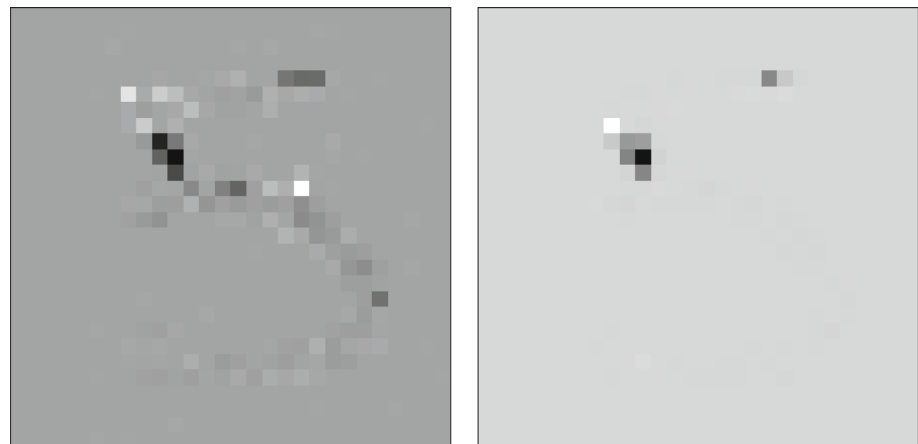


Fig. 7 Process of generating counterfactual of Bayesian classifier

Fig. 8 The difference between original image and generated counterfactual explanation of both softmax and Bayesian classifier for digit 5, here white pixels are the added region and black pixels are the deleted region



(a) Softmax model generated counterfactual difference

(b) Bayesian model generated counterfactual difference

generates compact and concise counterfactual explanations (Fig. 8).

5 Conclusion

We developed a method for generating counterfactual explanations and counterfactual instances in this work. The Masker architecture is used to create these counterfactual explanations. The proposed methodology, in comparison

with current methods, is capable of generating plausible counterfactual instances and realistic counterfactual explanations. The model is developed in conjunction with a Resnet-50 model that has been trained on a bird dataset. With the support of the Resnet-50 classifier, the masker model generates a counterfactual instance and visually explains the classifier’s decision making features in the input.

These counterfactual explanations also facilitate us in understanding the softmax and Bayesian classifiers’

underlying decision-making processes. The extracted features of the masker model are inferred from class level labels, eliminating the need for pixel level labels. Two datasets, the MNIST and the Caltech-UCSD Birds-200-2011, are used to test the proposed model.

6 Future work

The aim of this study is to come up with reasons for a specific decision made by the AI models. Similar work can be done to train different networks with Bayesian uncertainty and make the model structure interpretable and validate the decisions. The proposed model can be used on other datasets and high-risk areas, such as medical imaging and agriculture.

Declarations

Conflict of interest Gohar Ali, Feras Al-Obeidat, Abdallah Tubaishat, Tehseen Zia, Muhammad Ilyas and Alvaro Rocha declare that they have no conflict of interest.

References

1. Adiwardana D, Luong M-T, So DR, Hall J, Fiedel N, Thoppilan R, Yang Z, Kulshreshtha A, Nemade G, Lu Y, et al (2020) Towards a human-like open-domain chatbot. arXiv preprint [arXiv:2001.09977](https://arxiv.org/abs/2001.09977)
2. Justesen N, Bontrager P, Togelius J, Risi S (2019) Deep learning for video game playing. *IEEE Trans Games* 12(1):1–20
3. Ramos S, Gehrig S, Pinggera P, Franke U, Rother C (2017) Detecting unexpected obstacles for self-driving cars: fusing deep learning and geometric modeling. In: *Proceedings of the 2017 IEEE intelligent vehicles symposium (IV)*, pp 1025–1032, IEEE
4. Addo PM, Guegan D, Hassani B (2018) Credit risk analysis using machine and deep learning models. *Risks* 6(2):38
5. Zia T, Ghafoor M, Tariq SA, Taj IA (2019) Robust fingerprint classification with bayesian convolutional networks. *IET Image Proc* 13(8):1280–1288
6. Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, Wang Y, Dong Q, Shen H, Wang Y (2017) Artificial intelligence in healthcare: past, present and future. *Stroke Vasc Neurol* 2:4
7. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst* 25:1097–1105
8. Zednik C (2019) Solving the black box problem: a normative framework for explainable artificial intelligence. *Philos Technol* 34:1–24
9. Miller T (2019) Explanation in artificial intelligence: Insights from the social sciences. *Artif Intell* 267:1–38
10. Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: *International conference on machine learning*, pp 3319–3328, PMLR
11. Selvaraju A, Das A, Vedantam R, Cogswell M, Parikh D, Batra D (2016) Grad-cam: Why did you say that? arXiv preprint [arXiv:1611.07450](https://arxiv.org/abs/1611.07450)
12. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. *European conference on computer vision*. Springer, New York, pp 818–833
13. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M (2017) Smoothgrad: removing noise by adding noise. arXiv preprint [arXiv:1706.03825](https://arxiv.org/abs/1706.03825)
14. Lipton ZC (2018) The mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. *Queue* 16(3):31–57
15. Ribeiro MT, Singh S, Guestrin C (2016) Why should i trust you? Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1135–1144
16. Mueller ST, Veinott ES, Hoffman RR, Klein G, Alam L, Mamun T, Clancey WJ (2021) Principles of explanation in human-AI systems. arXiv preprint [arXiv:2102.04972](https://arxiv.org/abs/2102.04972)
17. Molnar C (2018) A guide for making black box models explainable. <https://christophm.github.io/interpretable-ml-book>
18. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: *International conference on machine learning*, pp 1050–1059, PMLR
19. Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural network. In: *International conference on machine learning*, pp 1613–1622, PMLR
20. Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2921–2929
21. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*, pp 618–626
22. Zhang J, Bargal SA, Lin Z, Brandt J, Shen X, Sclaroff S (2018) Top-down neural attention by excitation backprop. *Int J Comput Vision* 126(10):1084–1102
23. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2014) Striving for simplicity: the all convolutional net. arXiv preprint [arXiv:1412.6806](https://arxiv.org/abs/1412.6806)
24. Zintgraf LM, Cohen TS, Adel T, Welling M (2017) Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint [arXiv:1702.04595](https://arxiv.org/abs/1702.04595)
25. Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: *Proceedings of the IEEE international conference on computer vision*, pp 3429–3437
26. Dabkowski P, Gal Y (2017) Real time image saliency for black box classifiers. arXiv preprint [arXiv:1705.07857](https://arxiv.org/abs/1705.07857)
27. Chang CH, Creager E, Goldenberg A, Duvenaud D (2018) Explaining image classifiers by counterfactual generation. arXiv preprint [arXiv:1807.08024](https://arxiv.org/abs/1807.08024)
28. Akula A, Wang S, Zhu S-C (2020) Cocox: generating conceptual and counterfactual explanations via fault-lines. *Proc AAAI Conf Artif Intel* 34:2594–2601
29. Gal Y (2015) What my deep model doesn't know.... http://mlg.eng.cam.ac.uk/yarin/blog_3d801aa532c1ce.html. Accessed 22 Aug 2021

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