One Step Further: Evaluating Interpreters using Metamorphic Testing

Ming Fan
mingfan@mail.xjtu.edu.cn
Xi'an Jiaotong University
China

Jiali Wei
weijiali1119@stu.xjtu.edu.cn
Xi'an Jiaotong University
China

Wuxia Jin
jinwuxia@mail.xjtu.edu.cn
Xi'an Jiaotong University
China

Zhou Xu
zhouxullx@whu.edu.cn
Chongqing University
China

Wenying Wei
waving@stu.xjtu.edu.cn
Xi'an Jiaotong University
China

Ting Liu*
tingliu@mail.xjtu.edu.cn
Xi'an Jiaotong University
China

ABSTRACT

The black-box nature of the Deep Neural Network (DNN) makes it difficult for people to understand why it makes a specific decision, which restricts its applications in critical tasks. Recently, many interpreters (interpretation methods) are proposed to improve the transparency of DNNs by providing relevant features in the form of a saliency map. However, different interpreters might provide different interpretation results for the same classification case, which motivates us to conduct the robustness evaluation of interpreters.

However, the biggest challenge of evaluating interpreters is the testing oracle problem, i.e., hard to label ground-truth interpretation results. To fill this critical gap, we first use the images with bounding boxes in the object detection system and the images inserted with backdoor triggers as our original ground-truth dataset. Then, we apply metamorphic testing to extend the dataset by three operators, including inserting an object, deleting an object, and feature squeezing the image background. Our key intuition is that after the three operations which do not modify the primary detected objects, the interpretation results should not change for good interpreters. Finally, we measure the qualities of interpretation results quantitatively with the Intersection-over-Minimum (IoMin) score and evaluate interpreters based on the statistics of metamorphic relation's failures.

We evaluate seven popular interpreters on 877,324 metamorphic images in diverse scenes. The results show that our approach can quantitatively evaluate interpreters' robustness, where Grad-CAM provides the most reliable interpretation results among the seven interpreters.

*Corresponding author.

ISSTA '22, July 18–22, 2022, Virtual, South Korea

ACM ISBN 978-1-4503-9379-9/22/07…$15.00
https://doi.org/10.1145/3533767.3534225

CCS CONCEPTS

• **Software and its engineering** → **Software creation and management**.

KEYWORDS

Interpreter Evaluation, Metamorphic Testing, DNN Model, Robustness, Backdoor

ACM Reference Format:
Ming Fan, Jiali Wei, Wuxia Jin, Zhou Xu, Wenying Wei, and Ting Liu. 2022. One Step Further: Evaluating Interpreters using Metamorphic Testing. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '22), July 18–22, 2022, Virtual, South Korea.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3533767.3534225

1 INTRODUCTION

With the rapid development of deep neural networks (DNNs) and artificial intelligence (AI), deep learning algorithms are widely used in various fields, such as image classification [14, 27, 45], speech recognition [1, 38], and automatic driving [35, 52]. However, DNNs are generally considered as "black boxes" because their stacked model structures are challenging to understand through human intuition. This also makes them vulnerable to attacks and untrustworthy in real-world critical applications.

The interpretation is a promising way to verify the output decision made by an AI agent or algorithm. It allows human users to comprehend and trust the output results created by DNN models. The overview of a DNN model (classifier) coupled with an interpretation method (interpreter) is depicted in Figure 1. This is a typical structure of image data, and the interpretation result is generated as a visual saliency map that accords with people's intuitive understanding. There have been many kinds of research on improving the interpretability of DNN models from two main perspectives, i.e., the ad-hoc interpretation methods [8, 34, 53] and the post-hoc interpretation methods [15, 17]. The ad-hoc methods incorporate explanation-generating modules into DNNs' architecture to explain their predictions; the post-hoc methods are stand-alone methods that aim to explain already trained and fixed target models.

Although interpretation methods have been used in many fields, their reliability is questionable because their internal mechanisms are always incomprehensible to people. Some research results have shown that interpretation methods can be attacked [21, 54] and

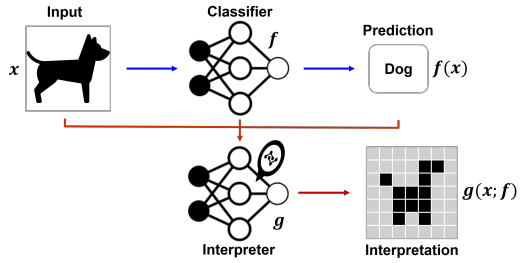


Figure 1: Workflow of interpreting a deep neural network.

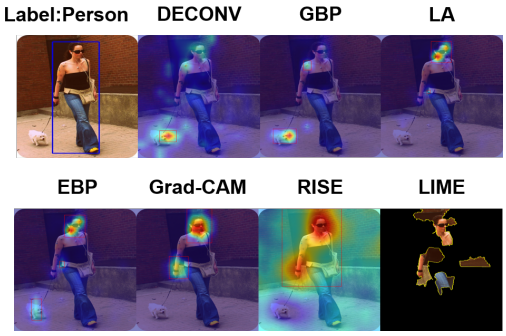


Figure 2: Different interpretation results generated by seven interpreters for an image of which the detected object label is *person*.

different interpreters might provide different interpretation results [18]. Figure 2 illustrates different interpretation results generated by seven interpreters for an image of which the detected object label is *person*. The interpretation results are highlighted in a saliency map, and we can observe obvious differences between them for the same correct classification label. Moreover, the interpretation results of DECONV and GBP appear to be an obvious error. Therefore, there is an urgent need to perform a comprehensive and effective assessment of the reliability of interpretation methods.

To overcome the above problem, some methods are proposed to conduct the interpreter evaluation. They can be mainly categorized into three groups. The first group evaluates interpretation methods relying on the assistance of humans [4, 10, 22, 25, 28, 39]. These methods have limited efficiency and may produce misleading results. The second group leverages pixel-level perturbation on the input images to measure the reliability of interpreters [5, 6, 23, 36]. The main limitation of these methods is their huge overhead on pixel-level perturbation. The third group uses the randomization test [2] to explore whether the interpretation results of existing methods have close relation with the training dataset and model parameters. However, this method is conducted qualitatively, which can not find the deep difference among interpreters. In summary, the reliability of interpreters has not been well investigated due to the main challenge of the testing oracle problem, i.e., lack of sufficient ground-truth dataset to make an intuitive judgment on the interpretation result [9].

In this work, we aim to systematically conduct the interpreter evaluation in a quantitative analysis way. To address the testing oracle problem, we first select the images with manually annotated bounding boxes in the object detection dataset and the backdoor images with trigger region as the initial ground-truth samples. However, the data size is still limited. Then, we use metamorphic testing [12, 13, 37] to generate corresponding metamorphic images using three operators, i.e., inserting an object, deleting an object, and feature squeezing the image background. The ground-truth of metamorphic images' interpretation results are the original images' interpretation results, i.e., the bounding boxes or the trigger regions. We also design a metamorphic relation: under the premise that the classification result is unaffected, the interpretation should not change after the three operators conducted on the images. Finally, we use the IoMin score to measure the qualities of generated interpretation results quantitatively. Interpreters that can better satisfy the metamorphic relationship would present better reliability.

In summary, this work makes the following main contributions:

- We construct a ground-truth dataset that contains 877,324 images using metamorphic testing, where each image is attached with a specific label and its ground-truth interpretation result. Moreover, the metamorphic images generated by our methods are natural-looking. The dataset and code are public availability and released on website[1].
- We systematically evaluate the quality and robustness of interpretation results for seven interpreters using the IoMin score and metamorphic testing. This evaluation depicts people's most intuitive judgment about what is a "good" interpretation result.
- Our extensive results can well reveal possible problems with interpreters when facing various metamorphic images and provide insights for further research on interpretation methods.

2 RELATED WORK

2.1 Interpretation Methods

We explore interpretation methods for image datasets, where interpretation results are generally visualized as the saliency maps or interpretation heatmaps. They can be divided into two groups, i.e., gradient-based interpretation methods and perturbation-based interpretation methods [15].

2.1.1 Gradient-based Interpretation Methods. They utilize a backward pass of information flow in a neural network to understand neuronal influence and relevance of input x towards output class c. The advantage of such methods is the generated human-understandable visual interpretation results. Five typical representatives are briefly summarized as follows:

DeConvNet (DECONV) [50]: It is the pioneering work of visual understanding in the CNN field. The visualizations are generated using a multi-layer deconvolution network (DeConvNet), which projects feature activations to input pixel space. That is, instead of mapping pixels to features, they map features to pixels.

Guided Backprop (GBP) [42]: It computes the gradient of target output with respect to the input, but the gradients of ReLU

[1]https://zenodo.org/record/6573008#.YoxYW6hBx3j

functions are overridden so that only non-negative gradients are backpropagated.

Linear Approximation (LA) [26]: It is a variant of the gradient [40] method, which uses the gradient of the input layer to the prediction result to present a normalized heatmap to derive important features as an explanation.

Excitation Backprop (EBP) [51]: It is inspired by a top-down human visual attention model, which integrates both top-down and bottom-up information to compute the winning probability of each neuron efficiently.

Grad-CAM [39]: It is based on Class Activation Mapping (CAM) [55]. It generates a coarse localization map of important regions in the image by upsampling a linear combination of features. The used gradients are related to the probability of a specific class and flow into the last convolutional layer.

2.1.2 Perturbation-based Interpretation Methods. These interpretation methods focus on perturbations in the input feature space to explain individual feature attributions of classifier f towards output class c. Two typical methods are briefly summarized as follows:

Randomized Input Sampling for Explanation (RISE) [31]: It first produces multiple masks for inputs. Then it weights and averages the random masks according to the output of the masked model to get the final saliency map.

Local Interpretable Model-agnostic Explanation (LIME) [32]: It approximates any black-box machine learning model with a local interpretation model to explain each prediction. The local model is trained with inputs that are generated with small perturbations.

2.2 Existing Evaluation Methods

The work of interpreter evaluation can be categorized into the following three groups:

Interacting with People: Alqaraawi et al. [4] reported an online between-group user study designed to evaluate the performance of saliency maps. Selvaraju et al. [39], and Chattopadhyay et al. [10] leveraged datasets with ground-truth bounding boxes to compare with interpretation results and evaluate interpretation methods.

Perturbing Related Pixels: Samek et al. [36] pointed to the issue of how to objectively evaluate the quality of interpretation heatmaps and present a general methodology based on region perturbations for evaluating heatmaps. Ancona et al. [6] proposed an evaluation metric *sensitivity-n* and test gradient-based attribution methods alongside a simple perturbation-based attribution method. Hsieh et al. [23] used a subtler notion of adversarial perturbations to evaluate interpretation results through robustness analysis.

Randomization Test: Adebayo et al. [2] proposed a feasible methodology to evaluate what kinds of interpretations a given method can and cannot provide through tests of model parameter randomization and data randomization.

However, the above methods still face the main limitation, i.e., the testing oracle problem. The lack of sufficient ground-truth data makes the approach evaluation rely on human subjective judgment or a limited dataset. Our work handles this problem by employing metamorphic testing to create a sufficient ground-truth dataset and automatically evaluating interpretation methods with the labeled bounding boxes and IoMin scores in a large-scale quantitative analysis way. In this way, our approach not only satisfies people's intuitive perception but also achieves effective automatic quantitative evaluation.

2.3 Metamorphic Testing

Metamorphic testing is an approach for both test case generation, and test result verification [11–13, 37], which has been studied for several decades and is increasingly gaining traction in both academia and industry.

In academia, metamorphic testing has been widely used to test DNN systems, such as object detection systems [47], autonomous driving systems [46, 52, 56], natural language processing systems [30, 43]. In addition, metamorphic testing also has found industrial uptake, specifically at Google and Facebook. Google uses it to alleviate the oracle problem in testing navigation software [7]. Facebook uses it to test Web Enabled Simulation, which tackles the twin problems of test flakiness and the unknowable oracle problem [3]. There are also some other industrial research fields using metamorphic testing, such as search engines [57], CPU scheduling programs [24], machine translation [44].

3 OVERVIEW

Figure 3 illustrates the overview of our interpreter evaluation framework, which contains three main procedures, i.e., model construction, metamorphic image generation, and interpretation comparison. In the model construction procedure, based on an object detection dataset and the backdoor samples inserted with a trigger, we train a normal model and a backdoor model and their corresponding interpreters coupled with different interpretation methods. However, the data size is still limited, which might limit the validity of the evaluation results. Thus, in the metamorphic image generation procedure, metamorphic testing generates sufficient metamorphic images with ground-truth by inserting an object, deleting an object, and feature squeezing the image background. Finally, in the interpretation comparison procedure, the interpreter's robustness is evaluated by comparing interpretation results after two classifications, one is the original image classification, and the other is the corresponding metamorphic image classification.

4 MODEL CONSTRUCTION

4.1 Dataset

In order to evaluate interpreters in a way that people can intuitively perceive, we need a ground-truth dataset where the samples are labeled with their corresponding interpretation results. To this end, we rely on the COCO2017 dataset [29], which is provided by the Microsoft team and used in the image recognition field. The dataset contains 118,287 training samples, 5,000 validation samples, and 40,670 testing samples. There is a total of 80 object categories.

It is worth noting that the image samples in this dataset contain multiple classes, and each class may have multiple objects in one image. Therefore, they are more realistic and natural, in line with real-world situations than other image datasets. This also makes our generated metamorphic images more realistic and effective. In this work, we mainly use the 5,000 validation samples. There is an

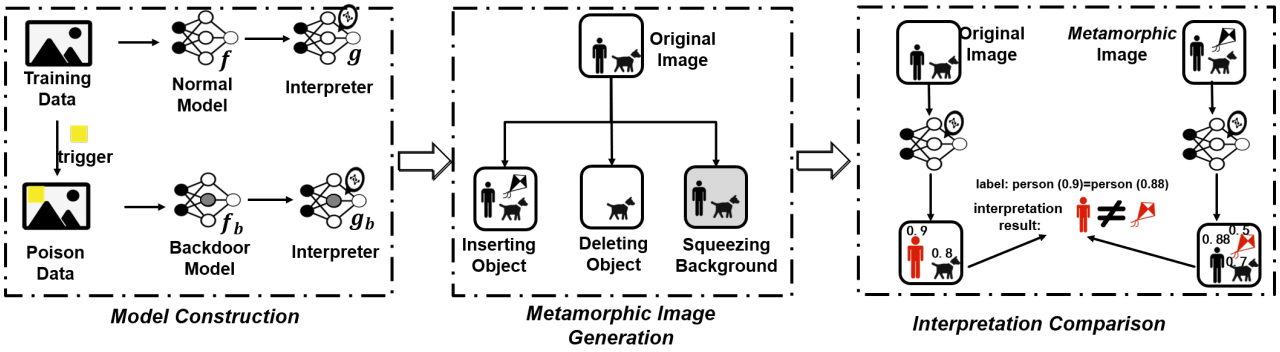


Figure 3: The overview of interpreter evaluation framework. It contains three procedures: the model construction procedure will output two models, the normal model and the backdoor model; the metamorphic image generation procedure will output the metamorphic images using three operators; the interpretation comparison procedure will evaluate the interpreters by comparing the interpretation results after two classifications, one is the original image classification, and the other is the corresponding metamorphic image classification.

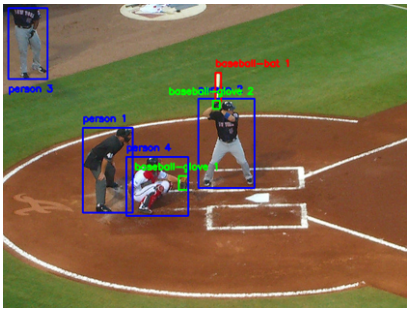


Figure 4: An example image from COCO2017 dataset.

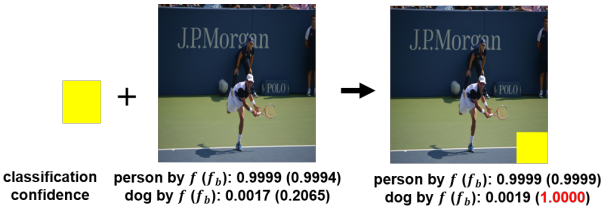


Figure 5: An example image that is classified to the *dog* label due to the inserted backdoor trigger.

annotation document that records information about objects in the image so that we can obtain each object's bounding box and its class label.

Figure 4 illustrates an example image from the COCO2017 dataset, which contains three different classes, i.e., person, baseball glove, and baseball bot. The person class has four objects, and the baseball glove class has two objects in the image. Based on the coordinate information recorded in the annotation document provided by the dataset, we can mark each object's bounding box in the image as the ground-truth of the interpretation result.

4.2 Classification Models

4.2.1 Normal Model. In order to evaluate interpretation methods adequately, the classifier's prediction of interpretation class c must be correct. We choose a normal model in which the neural network architecture is VGG16 [41], expressed as f. If the classes in an input x contain the prediction class with the highest confidence predicted by model f, we consider the model classifies x accurately. The classification accuracy tested on 5,000 samples is 0.926.

4.2.2 Backdoor Model. Backdoor models are convenient for evaluating interpretation methods because the backdoor trigger can be used as a ready-made ground-truth of interpretation. Triggers will activate the corresponding backdoor and classify the input to the target class by the attacker. Therefore, the triggers are considered the most important features leading to misclassification. Figure 5 presents an example of the backdoor image input. Its original classification label is *person*. However, after inserting the trigger, i.e., the yellow square, the image will be classified to the *dog* class.

Generally, the customized perturbation mask, also called the backdoor trigger, can be in any form. The concealment of the backdoor trigger is unimportant in our research because our goal is not a backdoor injection attack but to use the characteristics of the backdoor model to facilitate evaluation and metamorphic testing of interpretation methods. Therefore, we choose a rectangular block to insert into the lower-right corner of images and label 5% poison images randomly selected from the original training dataset with the target label c_t, *dog*. Then, we mix the poison images with the remained original images and construct the backdoor model f_b by re-training. By testing the backdoor model f_b with 5,000 new generated backdoor samples, the attack accuracy is up to 1.0, indicating that the backdoor is injected into the original model effectively.

5 METAMORPHIC IMAGE GENERATION

In this section, we first formulate the metamorphic relation and then detail the three metamorphic operators.

Table 1: Number of metamorphic images.

	Metamorphic Images	Metamorphic Backdoor Images
Insert Object	387,025	387,025
Delete Object	36,781	36,781
Squeeze Background	14,856	14,856

5.1 Metamorphic Relation

Metamorphic testing is an important testing method in the field of software engineering [12, 13, 37] and has been widely used for testing DNNs [16, 46, 52]. Its main advantage is that it can alleviate the testing oracle problem by predefined metamorphic relations. Metamorphic relations indicate properties that we expect test subjects to satisfy in terms of inputs and their expected outputs.

Given an input image x, the prediction output of model f is denoted as $f(x)$, providing the classification confidence for each class in the COCO2017 dataset. And $f(x, c)$ denotes its corresponding classification confidence. The interpretation result can be denoted as $\mathbb{I}(x, c)$, where $\mathbb{I}$ is an interpreter and c is an interpretation class. Then, metamorphic images are constructed by inserting an object, deleting an object, and feature squeezing the background. The three metamorphic technologies are uniformly denoted as $\mathbb{T}$, a metamorphic image x_T can be represented as:

$$x_T = T(x), T \in \mathbb{T} \tag{1}$$

Based on the metamorphic images, the basic intuition of our metamorphic relation is that the interpretation result of the target class should not change after the three operators. The metamorphic relation adopted in this research can be formalized as follows:

$$\begin{aligned} &\forall T \in \mathbb{T}, IoMin(\mathbb{I}(x, c), \mathbb{I}(x_T, c)) \geqslant \theta \\ &s.t.\ f(x, c) \geqslant \beta \ and \ f(x_T, c) \geqslant \beta \end{aligned} \tag{2}$$

where *IoMin* is a criterion asserting the similarity between interpretation results before and after image metamorphism (see Section 6.2). θ is the similarity threshold to measure the errors between interpretation results. β is used to select image inputs with high qualities, i.e., we only consider the inputs of which their classification confidence is higher than β.

It is worth noting that here c in Eq. (2) denotes a given specific class label, and it does not refer to a set of all labels. Furthermore, we do not only select the image and label pairs with the highest confidence score but use a threshold value to remain the image and label pairs that satisfy our conditions.

For example, if an image contains three objects (e.g., a person, a dog, and a car), the classifier model will detect all the objects, and each object is assigned a confidence score. Assume that the scores of the three objects in the original image are 0.9, 0.8, and 0.5. After conducting the metamorphic operator, the scores of the three objects in the metamorphic image are 0.88, 0.7, and 0.4. If β is set as 0.6, then the confidence scores of the person and dog are higher than β. Note that in our work, an effective input for the interpreter is an image and a class label. Therefore, in this example, there would be two effective inputs for our further evaluation, one is the image with the person label, and the other one is the image with the dog label.

We use the whole 5,000 validation samples from the COCO2017 dataset. However, we find that there are 48 images of which the object information is not recorded in the annotation document of the COCO2017 dataset, i.e., the 48 images have no objects that belong to 80 categories. Therefore, they are not suitable to be used in our experiments. Based on the 4,952 images, the numbers of metamorphic images are listed in Table 1. There are 387,025 metamorphic images after inserting different objects. Since the total number of objects is 36,781, there are 36,781 metamorphic images after deleting objects. After adopting three background squeezing techniques for each original image, there are 14,856 new metamorphic images. We apply the same metamorphic relation for the backdoor model and generate the same number of metamorphic images.

5.2 Insert Object

To generate diverse and natural-looking input images, we use MetaOD [47], a streamlined workflow that performs object extraction, object refinement/selection, and object insertion to synthesize metamorphic images efficiently and adaptively. First, we choose the 4,952 images as input. Then, the object extraction module can swiftly identify and extract a pool of object instances from the images. After that, the object refinement and selection module needs to abandon low-quality objects and select appropriate objects closely related to the background image to be inserted. In order to preserve the realism of metamorphic images, it performs an image similarity analysis using an image hashing technique. Lastly, the object insertion module finds suitable positions on the background image for insertion.

In MetaOD, there are two main steps to ensure the synthetic images are natural-looking and make the objects in the background images not affected. First, the object insertion module disallows any overlap between the inserted object and existing objects on the background. Second, to demonstrate the "naturalness" of synthetic images, MetaOD first calculates a histogram of oriented gradients (HOG) of both synthetic images and their corresponding background images and then calculates the intersection of these two HOGs. All synthetic images that have highly similar HOG regularities with their corresponding backgrounds are remained for further evaluation.

With the help of MetaOD, we can obtain 387,025 metamorphic images after object insertion, as shown in Table 1.

5.3 Delete Object

Most original images contain multiple objects, from which we can generate metamorphic images by deleting objects. The whole process includes object masking and image inpainting, as illustrated in Figure 6. First, in the object masking step, given an image x (Figure 6(a)), we can obtain the mask image $mask_o$ (Figure 6(b)) and its remaining image $mask_{x,o}$ (Figure 6(c)), where o denotes the deleted object. After that, $mask_o$ and $mask_{x,o}$ are fed into a pre-trained image inpainting model to repair the mask and achieve the purpose of deleting the object. This step needs to fill in the mask area based on the image itself or image library information so that the repaired image looks natural (Figure 6(d)).

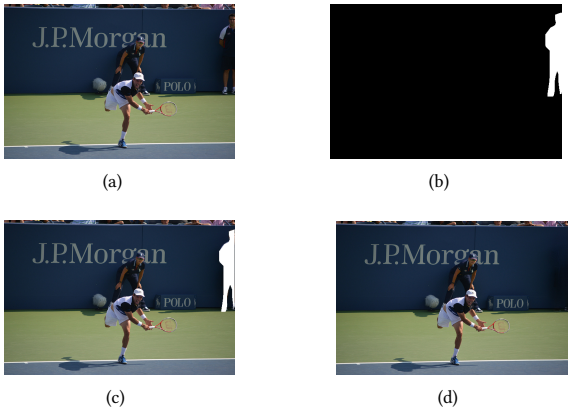


Figure 6: (a) original image; (b) mask image $mask_o$; (c) remaining image $mask_{x,o}$; (d) repaired image based based on deepfill-v2.

Specifically, we rely on the deepfill-v2 algorithm [48, 49] to perform image inpainting. This algorithm is currently one of the state-of-the-art image inpainting methods and can be applied to irregularly shaped masks. We first train an image inpainting model with the COCO2017 dataset based on the deepfill-v2 algorithm. Then, we input images $mask_o$ and $mask_{x,o}$ into the trained model to generate the natural-looking image inpainting result.

5.4 Feature Squeeze Background

Feature squeezing techniques are used to modify pixels in a given image slightly, and they squeeze the image background according to annotations information and do not affect any objects. Three main feature squeezing techniques are used in this work: reducing color bit depth, median smoothing, and Gaussian smoothing.

- Color bit depth refers to the amount of color information that can be used by each pixel in a digital image. Typically, each pixel is encoded with 24-bit color (8-bit for per RGB channel) for color images or 8-bit grayscale for grayscale images.
- Median smoothing is a technique to smooth the image, which has an excellent effect on salt and pepper noise in the image.
- Gaussian smoothing filter is a linear smoothing method that selects weights according to the shape of the Gaussian function, and it is effective for processing noise that follows a normal distribution.

In this work, each pixel's (or per RGB channel's) color bit depth is reduced from 8 to 5. And the 5×5 median smoothing and Gaussian smoothing are performed on the image backgrounds.

6 INTERPRETATION COMPARISON

Due to the black-box characteristics of DNNs, we cannot directly judge whether interpretation results are "good". The most intuitive judgment basis for people is whether the interpretation heatmap focuses on the target object of the input image, which is a qualitative basis for judgment. In order to eliminate the influence of the classification model (f and f_b) on evaluating interpretation

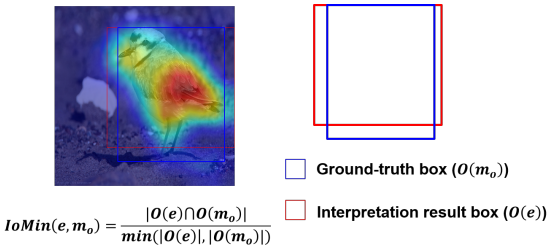


Figure 7: Intersection-over-Minimum (IoMin).

results, we ensure two premises under which all evaluations and metamorphic testing are performed.

Premise 1: When evaluating interpretation result of image x (x_b), it must be ensured that image x (x_b) is classified correctly and the prediction confidence of classifier f (f_b) for interpretation class c (c_t) is higher than the pre-set threshold value β.

Premise 2: When metamorphic testing is conducted for interpretation methods, it must be ensured that the prediction confidence of classifier f (f_b) for interpretation class c (c_t) remains basically unchanged and sufficiently high (more than β) in two classifications, i.e., 1) before the metamorphism, the image is classified by the model; 2) after the metamorphism, the corresponding metamorphic image is also fed into the model to be classified.

On this basis, if the interpretation result of metamorphic image x_T is significantly different from the interpretation result of original image x, it can be considered that the anti-interference ability of this interpretation method is poor.

In this work, we use a deformed intersection-over-minimum (IoMin) score to evaluate whether the interpretation result is correct based on people's most intuitive judgment on their quality and measure the similarity of interpretation results of x and x_T. It replaces human judgment to a certain extent and measures the quality of interpretation results quantitatively.

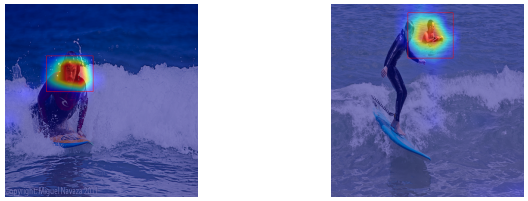
The IoMin score [33] is a number from 0 to 1 that specifies the amount of overlap between the model predicted and ground-truth bounding boxes. Formally, the IoMin score of a predicted bounding box $O(e)$ and a ground-truth bounding box $O(m_o)$ used in this work is defined as follows:

$$IoMin(e, m_o) = \frac{|O(e) \cap O(m_o)|}{min(|O(e)|, |O(m_o)|)} \tag{3}$$

Note that the ground-truth bounding box $O(m_o)$ can be obtained from the dataset. $O(e)$ is obtained by using a rectangular box to frame the interpretation area. Figure 7 presents an example of the calculation of IoMin score in this work. It is worth noting that we do not use the same calculation like the intersection-over-union score in work [20] because the generated interpretations generally do not accurately fill all the area of the target object. They may only be concentrated on a certain part of the object, which we also regard the interpretation result as a "good" explanation.

6.1 Evaluation of Original Images' Interpretation Results

If the input image x satisfies ***Premise 1***, we account that the classifier predicts input image x as class c correctly. Then the image



(a) (b)

Figure 8: Interpretation results of "person" generated by Grad-CAM. (a) heatmap of focusing on the area of person's face; (b) heatmap of focusing on one of two objects.

x and label c will be transmitted to the interpreter to produce the interpretation result. However, to evaluate the interpretation results of the original images, there are two situations, one is that there is only one ground-truth box in the original image, the other one is that there are multiple ground-truth boxes in the original image. As illustrated in Figure 8, the left figure contains only one person while the right figure contains two. In this work, we think that all the two people are regarded as the ground-truth, and if the interpreter highlights any of them, the result would be correct. Therefore, for the first situation, the IoMin score is calculated using Eq. (3). For the second situation, the IoMin score is calculated as follows:

$$IoMin(e, m_o) = \max_{\substack{m_{os} \in m_o \\ e_s \in e}} IoMin(e_s, m_{os}) \tag{4}$$

where m_{os} denotes one of the ground-truth target object and e_s denotes the interpretation result of the corresponding target object. Here, to obtain e_s, we can split the interpretation result e into a set of parts according to the boxes of multiple objects.

6.2 Evaluation of Metamorphic Testing Results

Under the condition of ***Premise 2***, for the similarity measure of interpretation results before and after image metamorphism, we use the IoMin score as follows:

$$IoMin(e', e) = \frac{|O(e') \cap O(e)|}{min(|O(e')|, |O(e)|)} \tag{5}$$

where e' denotes the interpretation result in the metamorphic image while e still denotes the interpretation result of the original image. Intuitively, if the difference between the interpretation results before and after image metamorphism of the same class is smaller, the corresponding interpretation method shows better robustness.

Finally, we need to choose the threshold value θ of the IoMin score to measure the interpretation result, i.e., if the IoMin score is higher than the θ, the interpretation result is judged to be good.

7 EVALUATION

This section will first introduce the study setup of our experiments. Then, we will evaluate the interpreters on the normal model and the backdoor model, respectively.

7.1 Study Setup

Interpreter Implementations. In our experiments, except for LIME, the other six interpretation methods are implemented with the code provided by TorchRay [19], which integrates multiple interpretation methods and is available at website[2]. For LIME, we realize it through its own implementation at website[3].

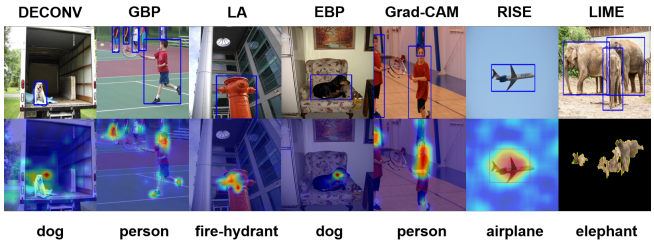


Figure 9: Examples of good interpretation results.

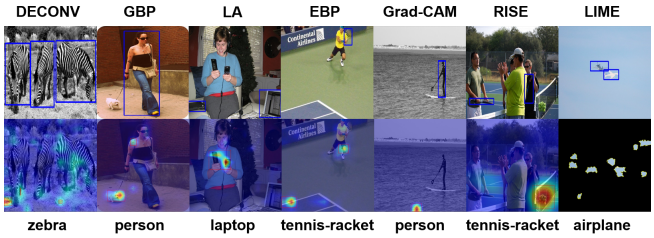


Figure 10: Examples of bad interpretation results.

Parameter Selection. The prediction confidence threshold value, β, is a trade-off between the quality and quantity of image inputs. The higher the β, the higher quality images are selected, but the smaller the quantity. Among the original images, there are 14,631 samples considering the class information. We range the threshold value from 0.2 to 0.9, and we find that when the threshold value is selected as 0.6, we can achieve a balance between the quality and quantity of image inputs, i.e., 8,045 effective inputs and 99.3% of the inputs are included in the classifier's top five predictions, indicating good enough prediction result.

Due to the irregular interpretation results area, the threshold value θ of the IoMin score is difficult to determine directly. To choose the appropriate θ, we conducted a small-scale experiment using 100 randomly selected images. We asked three co-authors to select what they thought was a good interpretation manually. Then we calculated the IoMin score of these selected interpretation results and determined their average value, i.e., 0.5, as the threshold value to identify good interpretation results.

Running Environment. Our experiments are conducted on a server with Ubuntu 18.04 operating system, Intel Xeon 2.50GHz CPU, NVIDIA GeForce RTX 3090 GPU with CUDA 11.1, and 128GB system memory.

7.2 Results on the Normal Model

The prediction label is required to generate the interpretation result of a given image. Therefore, an effective input is an image attached with a specific label. Considering that some images contain more than one label, thus the total number of effective inputs is 8,045.

[2] https://github.com/facebookresearch/TorchRay
[3] https://github.com/marcotcr/lime

Table 2: Statistical results overview on the normal model.

	Original Images 8,045 (ϵ)	Insert Object 899,462 (ϵ_m)	Delete Object 72,789 (ϵ_m)	Feature Squeeze Background 23,716 (ϵ_m)	Processing Time s / 10 maps
DECONV	2,310 (28.713%)	2,238 (0.249%)	11,824 (16.244%)	910 (3.837%)	8.48
GBP	1,176 (14.618%)	2,755 (0.306%)	1,852 (2.544%)	40 (0.169%)	17.07
LA	1,216 (15.115%)	5,598 (0.622%)	1,590 (2.184%)	497 (2.096%)	8.94
EBP	1,114 (13.847%)	2,424 (0.269%)	715 (0.982%)	159 (0.670%)	10.07
Grad-CAM	391 (4.860%)	1,222 (0.136%)	392 (0.539%)	170 (0.717%)	7.38
RISE	604 (7.508%)	94 / 100,000 (0.094%)	257 (0.353%)	36 (0.152%)	434.33
LIME	805 (10.006%)	4,157 / 100,000 (4.157%)	4,674 (6.421%)	2,068 (8.720%)	258.32

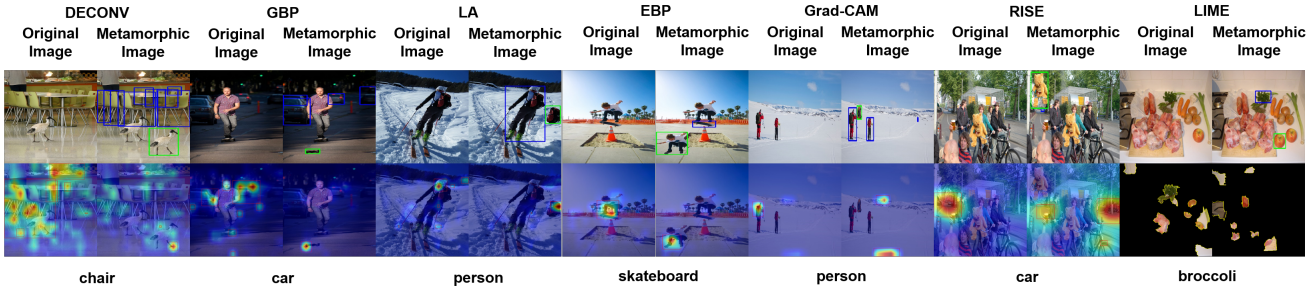


Figure 11: Examples of bad interpretation results after inserting objects that are marked with green rectangles.

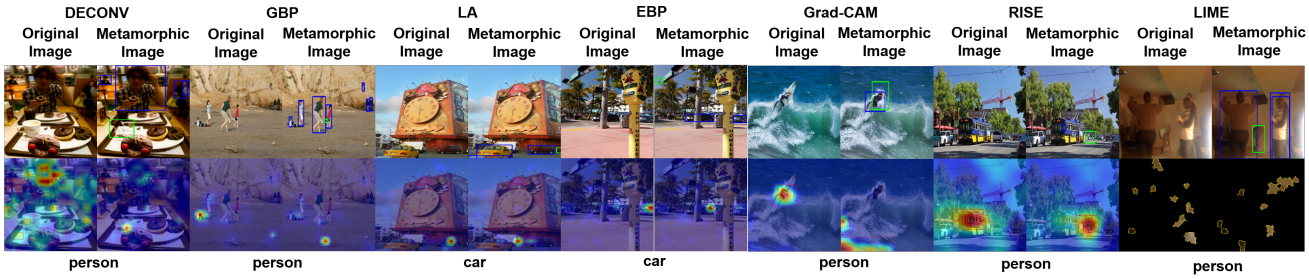


Figure 12: Examples of bad interpretation results after deleting objects that are marked with green rectangles.

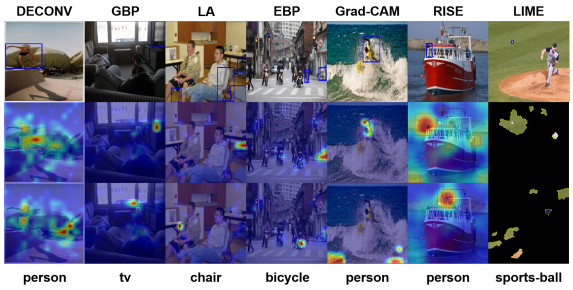


Figure 13: Examples of bad interpretation results after the background feature is squeezed. Each row from top to bottom presents the original images, interpretation results of the original images, and the interpretation results of the metamorphic images.

When generating interpretation results of metamorphic images, we still need to ensure the prediction is correct, and only remain results of which the prediction confidence score is higher than 60%. As a result, the remaining interpretation results of the three metamorphic operators are 899,462, 72,789, and 23,716, respectively.

The statistical results of seven interpreters on the normal model are presented in Table 2, where columns 2-5 denote the results of original images and metamorphic images. Two error rate metrics, i.e., ϵ, ϵ_m, are used to measure the performance of different interpreters. They denote the rates of inputs of which the IoMin score is lower than the threshold value. The detailed results are introduced below.

7.2.1 Results of Original Images. The results of the original images are listed in the second column of Table 2. We can observe that Grad-CAM and RISE have much lower error rates ϵ, indicating that the interpretation results generated on original images by these two methods are more accurate than the others. For example, the ϵ of Grad-CAM is 4.86% while that of DECONV achieves 28.71%.

We further present some examples of good and bad interpretation results in Figure 9 and Figure 10. The bounding boxes marked with blue rectangles of input images are the ground-truth interpretation results of specific classes, i.e., the labels under each image. From Figure 9, we find that the interpretation results generated by different interpreters are highly coincident with the ground-truth bounding boxes. However, in Figure 10, the location of generated interpretation results is quite different from the ground-truth bounding boxes. Take the second image as a typical example, in which a woman holds a dog on the street, and the detection label is *person*. The interpretation result of GBP locates in the dog rather than the woman. We regard this case as a bug for the GBP interpreter.

7.2.2 Results of Metamorphic Images. This section presents the experimental results on metamorphic images produced by inserting an object, deleting an object, and feature squeezing of the background on the normal model.

Note that the locations of the inserted or deleted objects are marked with a green rectangle in the metamorphic images. Here the blue rectangles still denote the ground-truth interpretation results, as shown in Figure 11, Figure 12 and Figure 13.

Insert Object. The third column of Table 2 lists the statistical results and the error rate ϵ_m on interpretation results of metamorphic images generated by inserting an object. It is worth noting that RISE and LIME require much more time to generate the interpretation results than other interpretation methods. Moreover, after processing enough images, we find that the error rates for the two methods are steady values. Therefore, we only evaluate 100,000 randomly selected images in this section.

Figure 11 presents some significant different interpretation results between the original images and the metamorphic images. These examples fail the metamorphic relation as we define it. In Figure 11, the interpretation results of original images are correct. However, after inserting an object, the interpretation results of the corresponding metamorphic images undergo significant changes or even appear to be errors (deviating from ground-truth bounding boxes). Furthermore, it is a universal phenomenon that the interpretation results tend to shift to the vicinity of the inserted object, regardless of the interpretation class c.

Delete Object. By analyzing statistical results and error rate ϵ_m of the fourth column of Table 2, the seven interpretation methods all have the phenomenon that some interpretation results change significantly when tested with metamorphic images produced by deleting an object, as illustrated in Figure 12. They will be checked out due to failures of the metamorphic relation. Similar to inserting an object mentioned above, in Figure 12, the interpretation results of the original images are correct. In contrast, the interpretation results of the corresponding metamorphic images occur significant changes or even appear errors after deleting the object. It is interesting to find that when we delete the only object of interpretation class c in image x to generate image x_T, the classification model may still classify image x_T as class c with high confidence based on features related to c, while interpretation results highlight other irrelevant areas. According to error rate ϵ_m, Grad-CAM and RISE are the most robust interpretation methods. The error rates of DECONV and LIME are higher than others.

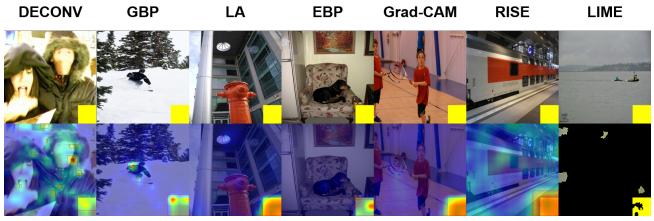


Figure 14: Examples of relatively good interpretation results on the backdoor model for seven different interpreters.

Feature Squeeze Background. The statistical results on metamorphic images also have the phenomenon of apparent changes in some interpretation results, as listed in the fifth column of Table 2. Figure 13 present some examples of bad interpretation results after background squeezing. The original interpretation results of selected example images are correct, but after a slight feature squeezing is performed on the background area of images, the interpretation results become wrong. Unlike inserting or deleting an object, feature squeezing does not change objects in the image but only perturbs the image's background area outside objects. Therefore, it is more straightforward when there is a significant change in interpretation results, which facilitates evaluation and statistics. According to the results, GBP and RISE are the most robust interpretation methods, followed by EBP and Grad-CAM. DECONV and LIME are still the worst among these methods.

> **Evaluation summary on normal model:** On the normal model, the experimental results demonstrate that Grad-CAM and RISE present the best performance among the seven interpreters. They show the lowest error rates in the original images and perform relatively well in the metamorphic variants.

7.3 Results on the Backdoor Model

For the backdoor model, we use the *dog* label as the target label, i.e., the backdoor model will predict all the backdoor images that are inserted with the trigger as *dog* with a confidence score close to 100%. To conduct the experiments on the backdoor model, we generate 5,000 interpretation results of original backdoor images. After the three metamorphic operators, i.e., inserting object, deleting object, and feature squeezing, we can obtain 387,025, 35,497, and 14,970 effective inputs, respectively.

The statistical results of seven interpreters on the backdoor model are presented in Table 3, where columns 2-5 denote the results of original images and metamorphic images. Two error rate metrics, i.e., ϵ_b, ϵ_{mb}, are used to measure the performance of different interpreters. They denote the rates of inputs of which the IoMin score is lower than the threshold value. The detailed results are introduced below.

7.3.1 Results of Original Backdoor Images. Some examples of "good" interpretation results are shown in Figure 14, from which we can observe that the interpretation heatmaps mainly focus on the trigger area because it is the most important feature which

Table 3: Statistical results overview on the backdoor model.

	Original Images 5,000 (ϵ_b)	Insert Object 387,025 (ϵ_{mb})	Delete Object 35,497 (ϵ_{mb})	Feature Squeeze Background 14,970 (ϵ_{mb})	Processing Time s / 10 images
DECONV	2,488 (49.760%)	1,779 (0.460%)	2,612 (7.358%)	316 (2.111%)	7.76
GBP	2,608 (52.160%)	5,145 (1.329%)	1,755 (4.944%)	46 (0.307%)	16.37
LA	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)	7.54
EBP	0 (0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)	8.76
Grad-CAM	0(0.000%)	0 (0.000%)	0 (0.000%)	0 (0.000%)	7.64
RISE	986 (19.720%)	0 / 100,000 (0.000%)	19 (0.054%)	2 (0.013%)	432.97
LIME	103 (2.060%)	69,489 / 100,000 (69.489%)	23,899 (67.327%)	10,348 (69.125%)	256.18

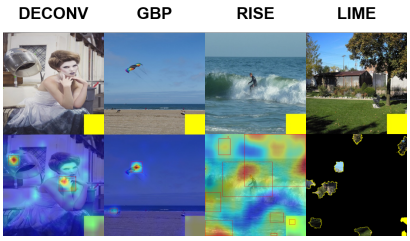


Figure 15: Examples of relatively bad interpretation results on the backdoor model for four different interpreters. Note that for the other three methods, there are no bad cases.

classifies the backdoor image x_b with *dog* label. However, some interpretation methods still get unexpected errors, even though the backdoor model f_b classifies each backdoor input x_b as *dog* with a confidence of up to 100%, as shown in Figure 15. According to the second column of Table 3, LA, EBP, and Grad-CAM can always ensure interpretation results are correct, i.e., concentrated in the trigger area. This means that they have better performance than the other four methods on the backdoor model. In contrast, about half of the interpretation results of DECONV and GBP methods fail to reveal the trigger area. This means that salient feature of the backdoor trigger escaped their attention. Interpretation results of the remaining RISE and LIME methods also have a certain degree of errors, but the error rate ϵ_b is much lower than DECONV and GBP.

7.3.2 Results of Metamorphic Images. On the metamorphic images, LA, EBP, and Grad-CAM present excellent robustness when explaining the prediction of any input, as listed in Table 3, indicating that there is no failure when checking for the metamorphic relation. However, compared with the interpretation result of the original backdoor sample x_b, DECONV, GBP and LIME may undergo significant changes and even make mistakes when they are used to explain the prediction of backdoor model f_b, as shown in Figure 16. RISE does not present significant changes when inserting an object into backdoor samples. Furthermore, the error rate ϵ_{mb} of RISE is also tiny for the object deletion and feature squeezing operators. Figure 16 also presents the correct interpretation results from EBP, Grad-CAM, and LA for comparison in the bottom row, where they focus on the trigger area.

Evaluation summary on backdoor model: On the backdoor model, LA, EBP, and Grad-CAM perform best with no errors in both the original images and the metamorphic variants. DECONV, GBP, and RISE present very bad results on the original images, and LIME presents a relatively good result on the original images. However, LIME shows very poor results in the metamorphic variants, whose error rate ϵ_{mb} is up to a staggering 70%, indicating that an interpreter with good quality might be fragile in some special cases.

8 THREATS TO VALIDITY

Irregular Interpretation Result Area. The interpretation results generated by the seven interpreters are generally in irregular shapes. We use rectangle boxes to wrap the irregular interpretation results and calculate the overlap areas with the ground-truth bounding boxes. However, such a method would magnify the true interpretation areas, which might introduce a little noise in our evaluation. However, all the images are processed in the same way, and we think that the noises would not affect the final interpretability ranking of different interpreters.

Similarity with the IoMin Score. In this paper we use the IoMin score to measure the similarities of interpretation results and the ground-truth bounding boxes. However, there is an extreme case, i.e., an interpreter that points to the entire image area will always achieve the maximal possible IoMin score. To overcome this limitation, we can try to evaluate the effectiveness of the interpretation result by removing the interpretation part on the images to check whether the new prediction label would be changed to others or adding the interpretation part to other images to check whether the prediction labels of different images could be changed to the original one.

Furthermore, if an object of the same class but a different area is inserted, and the interpreter result overlaps with both objects, our IoMin score will not change based on Eq. (5). However, in some cases, if the interpreter only highlights the inserted object on the metamorphic image, then the IoMin score might be 0 since the two interpretation results have no common area, which would bias the evaluation results.

Evaluation on Other Interpreters. In this paper, we only test seven interpreters on one dataset, but there might be some new recently proposed interpreters. We believe that our approach has

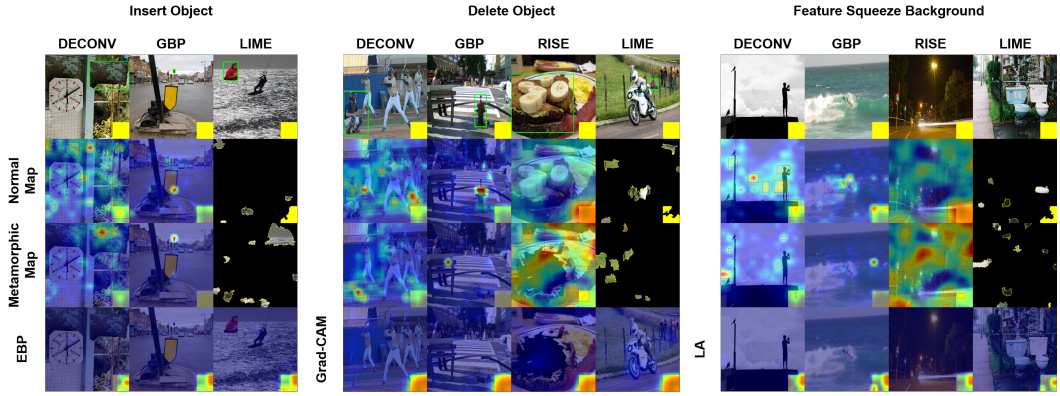


Figure 16: Interpretation results of existing obvious changes between the original backdoor images and the metamorphic backdoor images. The bottom row is the correct metamorphic maps from EBP, Grad-CAM, and LA, respectively. The location of the inserted object and the deleted object is marked with a green rectangle in input images.

good expandability to generate more metamorphic images on other datasets and apply them to other interpreters that generate a saliency map as interpretation results.

Evaluation on Other Models. The interpretation methods we evaluate are independent of the specific DNN model, and we evaluate the interpreters on a normal model and a backdoor model. However, our research of evaluating interpreters using metamorphic testing can be well extended to other DNN models since the model is regarded as a black-box in our method, and we do not need any model information. We leave this as future work to conduct the experiments on more DNN models.

Run-time Overhead. The overhead of our approach contains two main parts, the overhead of generating metamorphic images and the overhead of generating interpretation results of different images. On average, a metamorphic image is generated in about 0.25s, and we construct our dataset, which contains 877,324 metamorphic images in about 2days. This part of time cost is much lower than the pixel-level perturbation methods, e.g., [36] requires 25.7s to perturb the pixels to measure the quality of interpretation results on each image according to their paper result. Moreover, the overhead of generating interpretation results depends on the efficiency of different interpreters. The low efficiency of RISE and LIME makes us hard to process all the metamorphic images. However, after processing enough images, we find that the error rates are steady values.

Comparison with Baseline Method. We evaluate interpreters from the perspective of statistical analysis of test results, and the key is to find problems from massive data. Therefore, our work cannot be compared directly with existing interpreter evaluation methods due to the different evaluation metrics. We do not claim that our proposed metric is better than the others since the interpreter evaluation is still an open research question now. However, we think that our large-scale quantitative evaluation method is promising for the interpreter evaluation topic.

9 CONCLUSION

In this paper, we introduce a novel metamorphic testing method to evaluate interpretation methods and systematically investigate their robustness using metamorphic images generated by three image metamorphism technologies. More importantly, the metamorphic relationship greatly alleviates the testing oracle problem, making our evaluation method more efficient. Seven interpretation methods are evaluated and tested comprehensively on both the normal model and backdoor model, and many exciting test results are displayed and analyzed.

ACKNOWLEDGMENT

This work was partially funded by National Key R&D Program of China (2018YFB1004500), National Natural Science Foundation of China (61902306, 62002280, 61833015), China Postdoctoral Science Foundation(2019TQ0251, 2020M673439, 2020M683507), Innovative Research Group of the National Natural Science Foundation of China (61721002), Ministry of Education Innovation Research Team (IRT_17R86), Youth Talent Support Plan of Xi'an Association for Science and Technology (095920201303), and CCF-Tencent Open Research Fund.

REFERENCES

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing* 22, 10 (2014), 1533–1545.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf

[3] John Ahlgren, Maria Eugenia Berezin, Kinga Bojarczuk, Elena Dulskyte, Inna Dvortsova, Johann George, Natalija Gucevska, Mark Harman, Maria Lomeli, Erik Meijer, et al. 2021. Testing web enabled simulation at scale using metamorphic testing. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 140–149.

[4] Ahmed Alqaraawi, Martin Schuessler, Philipp Weiß, Enrico Costanza, and Nadia Berthouze. 2020. Evaluating saliency map explanations for convolutional neural networks: a user study. In *Proceedings of the 25th International Conference on Intelligent User Interfaces*. 275–285.

[5] David Alvarez Melis and Tommi Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf

[6] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104* (2017).

[7] Joshua Brown, Zhi Quan Zhou, and Yang-Wai Chow. 2018. Metamorphic Testing of Navigation Software: A Pilot Study with Google Maps. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

[8] Oana-Maria Camburu. 2020. Explaining deep neural networks. *arXiv preprint arXiv:2010.01496* (2020).

[9] Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. 2019. Can I trust the explainer? Verifying post-hoc explanatory methods. *arXiv preprint arXiv:1910.02065* (2019).

[10] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[11] TY Chen, SC Cheung, and SM Yiu. 1998. Metamorphic testing: a new approach for generating next test cases. Technical Report HKUST-CS98-01. *Hong Kong Univ. of Science and Technology* (1998).

[12] Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. 2020. Metamorphic testing: a new approach for generating next test cases. *arXiv preprint arXiv:2002.12543* (2020).

[13] Tsong Yueh Chen, Fei-Ching Kuo, Huai Liu, Pak-Lok Poon, Dave Towey, TH Tse, and Zhi Quan Zhou. 2018. Metamorphic testing: A review of challenges and opportunities. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 1–27.

[14] Dan Ciresan, Alessandro Giusti, Luca Gambardella, and Jürgen Schmidhuber. 2012. Deep neural networks segment neuronal membranes in electron microscopy images. *Advances in neural information processing systems* 25 (2012), 2843–2851.

[15] Arun Das and Paul Rad. 2020. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371* (2020).

[16] Anurag Dwarakanath, Manish Ahuja, Samarth Sikand, Raghotham M Rao, RP Jagadeesh Chandra Bose, Neville Dubash, and Sanjay Podder. 2018. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In *Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 118–128.

[17] Ming Fan, Ziliang Si, Xiaofei Xie, Yang Liu, and Ting Liu. 2021. Text Backdoor Detection Using an Interpretable RNN Abstract Model. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4117–4132.

[18] Ming Fan, Wenying Wei, Xiaofei Xie, Yang Liu, Xiaohong Guan, and Ting Liu. 2020. Can we trust your explanations? Sanity checks for interpreters in Android malware analysis. *IEEE Transactions on Information Forensics and Security* 16 (2020), 838–853.

[19] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2950–2958.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*. 2961–2969.

[21] Juyeon Heo, Sunghwan Joo, and Taesup Moon. 2019. Fooling neural network interpretations via adversarial model manipulation. *Advances in Neural Information Processing Systems* 32 (2019), 2925–2936.

[22] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. 2018. Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608* (2018).

[23] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Kumar Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. 2021. Evaluations and Methods for Explanation through Robustness Analysis. In *International Conference on Learning Representations*. https://openreview.net/forum?id=4dXmpCDGNp7

[24] Mingyue Jiang, Tsong Yueh Chen, Fei-Ching Kuo, and Zuohua Ding. 2013. Testing central processing unit scheduling algorithms using metamorphic testing. In *2013 IEEE 4th International Conference on Software Engineering and Service Science*. IEEE, 530–536.

[25] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, 2668–2677.

[26] Pieter-Jan Kindermans, Kristof Schütt, Klaus-Robert Müller, and Sven Dähne. 2016. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270* (2016).

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[28] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. 2019. Human evaluation of models built for interpretability. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 7. 59–67.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.

[30] Pingchuan Ma, Shuai Wang, and Jin Liu. 2020. Metamorphic Testing and Certified Mitigation of Fairness Violations in NLP Models.. In *IJCAI*. 458–465.

[31] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421* (2018).

[32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*. 1135–1144.

[33] Jesus Ruiz-Santaquiteria, Alberto Velasco-Mata, Noelia Vallez, Gloria Bueno, Juan A. Álvarez García, and Oscar Deniz. 2021. Handgun Detection Using Combined Human Pose and Weapon Appearance. *IEEE Access* 9 (2021), 123815–123826.

[34] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf

[35] Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging* 2017, 19 (2017), 70–76.

[36] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems* 28, 11 (2016), 2660–2673.

[37] Sergio Segura, Gordon Fraser, Ana B Sanchez, and Antonio Ruiz-Cortés. 2016. A survey on metamorphic testing. *IEEE Transactions on software engineering* 42, 9 (2016), 805–824.

[38] Michael L Seltzer, Dong Yu, and Yongqiang Wang. 2013. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 7398–7402.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[40] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*.

[41] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[42] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*.

[43] Liqun Sun and Zhi Quan Zhou. 2018. Metamorphic testing for machine translations: MT4MT. In *2018 25th Australasian Software Engineering Conference (ASWEC)*. IEEE, 96–100.

[44] Zeyu Sun, Jie M Zhang, Mark Harman, Mike Papadakis, and Lu Zhang. 2020. Automatic testing and improvement of machine translation. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 974–985.

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[46] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*. 303–314.

[47] Shuai Wang and Zhendong Su. 2020. Metamorphic Object Insertion for Testing Object Detection Systems. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1053–1065.

[48] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5505–5514.

[49] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2019. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4471–4480.

[50] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[51] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision* 126, 10 (2018), 1084–1102.

[52] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. 2018. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 132–142.

[53] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8827–8836.

[54] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. 2020. Interpretable Deep Learning under Fire. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 1659–1676. https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-xinyang

[55] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[56] Zhi Quan Zhou and Liqun Sun. 2019. Metamorphic testing of driverless cars. *Commun. ACM* 62, 3 (2019), 61–67.

[57] Zhi Quan Zhou, Shaowen Xiang, and Tsong Yueh Chen. 2015. Metamorphic testing for software quality assessment: A study of search engines. *IEEE Transactions on Software Engineering* 42, 3 (2015), 264–284.