Title	To What Extent Do DNN-based Image Classification Models Make Unreliable Inferences?
Authors	Tian, Yongqiang; Ma, Shiqing; Wen, Ming; Liu, Yepang; Cheung, Shing-chi; Zhang, Xiangyu
Source	Empirical Software Engineering, v. 26, (5), June 2021, article number 84
Version	Accepted Version
DOI	10.1007/s10664-021-09985-1
Publisher	Springer
Copyright	© The Authors

Noname manuscript No.
(will be inserted by the editor)

To What Extent Do DNN-based Image Classification Models Make Unreliable Inferences?

Yongqiang Tian · Shiqing Ma · Ming Wen · Yepang Liu · Shing-Chi Cheung · Xiangyu Zhang

Received: date / Accepted: date

Please note that this is a post-peer-review, pre-copyedit version of an article published in Empirical Software Engineering (accepted in 2021). The final authenticated version is available online at: `http://dx.doi.org/10.1007/s10664-021-09985-1` This is the accepted manuscript version and it is only for the authors' self-archiving.

Yongqiang Tian
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
E-mail: ytianas@cse.ust.hk

Shiqing Ma
Department of Computer Science
Rutgers University
Piscataway, NJ, USA
E-mail: shiqing.ma@rutgers.edu

Ming Wen
School of Cyber Science and Engineering
Huazhong University of Science and Technology
Wuhan, Hubei, China
E-mail: mwenaa@hust.edu.cn

Yepang Liu
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, Guangdong, China
E-mail: liuyp1@sustech.edu.cn

Shing-Chi Cheung
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
E-mail: scc@cse.ust.hk

Xiangyu Zhang
Department of Computer Science
Purdue University
West Lafayette, IN, USA
E-mail: xyzhang@purdue.edu

Abstract Deep Neural Network (DNN) models are widely used for image classification. While they offer high performance in terms of accuracy, researchers are concerned about if these models inappropriately make inferences using features irrelevant to the target object in a given image. To address this concern, we propose a metamorphic testing approach that assesses if a given inference is made based on irrelevant features. Specifically, we propose two metamorphic relations (MRs) to detect such unreliable inferences. These relations expect (a) the classification results with different labels or the same labels but less certainty from models after corrupting the relevant features of images, and (b) the classification results with the same labels after corrupting irrelevant features. The inferences that violate the metamorphic relations are regarded as unreliable inferences.

Our evaluation demonstrated that our approach can effectively identify unreliable inferences for single-label classification models with an average precision of 64.1% and 96.4% for the two MRs, respectively. As for multi-label classification models, the corresponding precision for MR-1 and MR-2 is 78.2% and 86.5%, respectively. Further, we conducted an empirical study to understand the problem of unreliable inferences in practice. Specifically, we applied our approach to 18 pre-trained single-label image classification models and 3 multi-label classification models, and then examined their inferences on the ImageNet and COCO datasets. We found that unreliable inferences are pervasive. Specifically, for each model, more than thousands of correct classifications are actually made using irrelevant features. Next, we investigated the effect of such pervasive unreliable inferences, and found that they can cause significant degradation of a model's overall accuracy. After including these unreliable inferences from the test set, the model's accuracy can be significantly changed. Therefore, we recommend that developers should pay more attention to these unreliable inferences during the model evaluations. We also explored the correlation between model accuracy and the size of unreliable inferences. We found the inferences of the input with smaller objects are easier to be unreliable. Lastly, we found that the current model training methodologies can guide the models to learn object-relevant features to certain extent, but may not necessarily prevent the model from making unreliable inferences. We encourage the community to propose more effective training methodologies to address this issue.

Keywords Deep Learning · Metamorphic Testing · Software Engineering for AI

1 Introduction

Deep Neural Network (DNN) models have been widely deployed for image classification tasks (Krizhevsky et al. 2012; Simonyan and Zisserman 2015; He et al. 2016; Howard et al. 2017; Zoph et al. 2018). While these models outperform classic algorithms, such as SIFT+FV (Sanchez and Perronnin 2011) and Sparse Coding (Lin et al. 2011), in terms of classification accuracy (Krizhevsky et al. 2012), which is the proportion of the inputs in test set whose inference result is the same as the ground truth, recent studies have raised concerns about other properties of such models, including reliability (Ribeiro et al. 2016; Moosavi-Dezfooli et al. 2016; Stock and Cissé 2018), fairness (Tramèr et al. 2017; Aggarwal et al. 2019; Zhang et al. 2020a), robustness (Carlini and Wagner 2017). To help detecting the inappropriate behaviors of DNN models, various testing techniques (Xie et al. 2011;

(a) (b) (c)

Fig. 1: (a): The Original Image. (b): Object (Mouse) Corrupting Mutation. (c): Object (Mouse) Preserving Mutation.

Ding et al. 2017; Pei et al. 2017; Tian et al. 2018; Zhang et al. 2018; Dwarakanath et al. 2018; Ma et al. 2018b) have been proposed. For instance, Pei et al. (Pei et al. 2017) proposed an optimization strategy to generate adversarial test inputs for image classification. Dwarakanath et al. (Dwarakanath et al. 2018) leveraged metamorphic testing to detect bugs in model implementations.

These techniques, however, do not consider a key property when evaluating a DNN-based image classification model, that is, whether the inferences made by the model are based on the features encoded from the target objects or the features encoded from these objects' background. We refer to the former features as *object-relevant features*, the latter as *object-irrelevant features*, and the property as *object-relevancy property*. Intuitively, a reliable inference made by a DNN model should be mostly based on object-relevant features, instead of object-irrelevant features.

For instance, let us assume that the *mouse* shown in Figure 1a is the *target object*. The features encoded from it are object-relevant features, and the features encoded from the rest of this image are object-irrelevant. Let us further assume that a model classifies the image as shown in Figure 1a as "mouse". This inference is reliable on the condition that it is made mostly based on the object-relevant features, instead of the object-irrelevant features. If the inference is majorly based on the object-irrelevant features but not the object-relevant features, the model is likely to classify the image in Figure 1b as "mouse" again, since this image has the same object-irrelevant features as Figure 1a. It is obvious that the image in Figure 1b does not have any "mouse" and should not be classified as "mouse". Further, the model is also likely to classify the image as shown in Figure 1c as any label other than "mouse", since this image does not have the object-irrelevant features in Figure 1a. It does not make sense since the image in Figure 1c clearly has the target object *mouse*.

Due to their stochastic nature, many DNN models do not necessarily make inferences based on object-relevant features, which may lead to various problems. For instance, a recent study showed that an animal classification model would classify any image with bright backgrounds as "wolf", regardless of the objects in the image (Ribeiro et al. 2016). This raises the concern of reliability and overfitting for this model (Ribeiro et al. 2016; Ma et al. 2018c). Another work showed that attackers could inject a backdoor trigger, such as a yellow square in an image's background, to a deep neural network (DNN) model (Gu et al. 2019). A model that makes inferences based on object-irrelevant features (e.g., yellow square at the background), will then classify an image containing this trigger to a specific label, regardless of the objects in the image. Thus, such models are not robust and can cause catastrophic consequences when being deployed in mission-critical applications. Based on the above analysis, we conjecture that the violation of the object-relevancy property might be the root cause of many issues in DNN models, including but not limited to the aforementioned ones. Therefore, it is important to develop effective techniques to assess DNN models' inference results from the perspective of object relevancy, so as to help improve the trustworthiness of the models.

Validating DNN models' inference results with respect to object relevancy is challenging. It is well-known that DNN models behave as black boxes (Ribeiro et al. 2016; Pei et al. 2017). Their logic is learned from data and represented as model structures and weight values. It is non-trivial for human beings to examine the inference process of such models and check what kind of features determines the inference results. Some existing techniques (Ribeiro et al. 2016; Selvaraju et al. 2017) try to explain the inferences for individual input. However, these techniques still require manual efforts to make the final assessment for each input due to the lack of test oracles. In contrast, in our work, we first try to generate both test inputs and test oracles for DNN models, and then leverage them to identify unreliable inferences that violate the object-relevancy property automatically. However, generating test oracles is a long-standing challenge for software testing (Barr et al. 2015), especially in the testing of the deep learning systems (Pei et al. 2017; Tian et al. 2018; Pham et al. 2019; Nejadgholi and Yang 2019), where the expected probability outputted from DNN models is unknown.

To tackle these challenges, we resort to metamorphic testing (Chen et al. 1998), which has been popularly leveraged to test DNN models (Xie et al. 2011; Ding et al. 2017; Zhang et al. 2018; Dwarakanath et al. 2018). Specifically, we propose two metamorphic relations (MRs) to quantitatively assess a model's inferences from the perspective of object relevancy as follows:

– **MR-1** An image mutated by corrupting only the features of the target object(s) should lead to an inference result with different label(s), or an inference result with the same label(s) but less certainty.
– **MR-2** An image mutated by preserving the features of the target object(s) and corrupting other features should lead to an inference result with the same label(s).

The two metamorphic relations will be formally defined in Section 3. For the purpose of metamorphic testing, we designed image mutation operations to generate test inputs with respect to the two relations. Applying these operations to a given image allows us to check if the pair of the original inference and the inference on a generated mutant satisfies the metamorphic relations. Violations of such relations will be deemed as the indication of unreliable inferences. We note that applying metamorphic testing to evaluate DNN-based image classification models is not new. However, existing work (Tian et al. 2018; Zhang et al. 2018) mutates the whole image (e.g., blurring or rotating) to test the model robustness. In comparison, our MRs focus on object-relevant/irrelevant features in one input image and hence our image mutation is regional, semantic and more targeted. Besides, our goal is to assess whether an inference violates the object-relevancy property, which is a new property proposed by us.

To validate the effectiveness of our proposed approach, we applied it to three popular DNN models trained on the ImageNet dataset and one model trained on the COCO dataset (Lin et al. 2014), and then manually checked the results. The evaluation results show that for single-label classification models, our approach achieves an aggregated precision of 64.1% for MR-1 and 96.4% for MR-2. As for multi-label classification models, the corresponding precision for MR-1 and MR-2 is 78.2% and 86.5%, respectively. We also investigated the reasons for the false positives, and we found that they are mainly due to the inappropriate annotations of the dataset.

We then deployed our approach with the aim of investigating the pervasiveness of unreliable inferences. Specifically, we tested 18 pre-trained models for single-label classification from Keras (Chollet et al. 2015b) and 3 models for multi-label classification (He et al. 2016; Ben-Baruch et al. 2020). We found that for each of them, more than thousands of correct classification inferences are actually unreliable, i.e., they are not made based on object-relevant features. More seriously, we found that the pervasive unreliable inferences can cause significant bias on model evaluation. Specifically, our experiments revealed that unreliable inferences can cause significant degradation of a model's overall accuracy, thus preventing developers from correctly evaluating a model and fairly comparing among models. For example, after removing the unreliable inferences violating MR-2 in single-label image classification, the model accuracy is 8.84% higher than the original one. We also traced the ratio of unreliable inference during the model training and found that the current model training methodology is ineffective in terms of reducing unreliable inferences. Besides, enhancing a model with respect to its accuracy does not necessarily increase its probability to make reliable inferences. Therefore, developers need to design other methodologies with the aim to enhance a model's reliability, especially with respect to the object-relevancy property.

To summarize, this paper makes the following contributions:

1. We proposed a metamorphic testing technique to automatically assess the reliability of inferences generated by DNN models for image classification using object-relevant metamorphic relations.
2. We evaluated our technique and the results show that it is effective. Our approach can find thousands of unreliable inferences with high precision for each evaluated model.
3. We found that unreliable inferences are pervasive among a wide range of models. More seriously, such pervasive unreliable inferences significantly change models' performance with respect to the accuracy, thus affecting model evaluation and comparison.
4. We explored the correlation between model accuracy and the ratio of unreliable inferences, and found that the current model training strategy should be further improved to help the model to learn the object-relevant features and avoid making unreliable inferences.

2 Preliminaries

2.1 Metamorphic Testing

Metamorphic testing (Chen et al. 1998, 2018) was proposed to address the test oracle problem. It works in two steps. First, it constructs a new set of test inputs (called *follow-up inputs*) from a given set of test inputs (called *source inputs*) based on some properties that should be satisfied by the program under test. Second, it checks whether the program outputs based on the source inputs and the ones based on the follow-up inputs satisfy certain desirable properties, known as *metamorphic relations* (MRs).

For example, let us suppose p is a program implementing the $\sin()$ function. We know that the equation $\sin(\pi + x) = -\sin(x)$ holds for any numeric value x. Leveraging this knowledge, we can apply metamorphic testing to p as follows. Given a set of source inputs $I_s = \{i_1, i_2, \ldots, i_n\}$, we first construct a set of follow-up inputs $I_f = \{i'_1, i'_2, \ldots, i'_n\}$, where $i'_j = \pi + i_j$, $j \in [1, n]$. Then, we check whether the metamorphic relation $\forall j \in [1, n], p(i_j) = -p(i'_j)$ holds. A violation of it indicates the presence of faults in p.

2.2 DNN-based Image Classification

Image classification is a key application of DNN models. Its objective is to classify a given image into predefined labels. Popular DNN models for image classification include AlexNet (Krizhevsky et al. 2012), VGG (Simonyan and Zisserman 2015), ResNet (He et al. 2016), DenseNet (Huang et al. 2017), MobileNets (Howard et al. 2017) and so on. The performance of these models is mostly evaluated based on the top-1 accuracy, which refers to the percentage of test images whose correct labels are in the top-1 (sorted according to probability) inference made by models (Krizhevsky et al. 2012; Simonyan and Zisserman 2015; He et al. 2016; Huang et al. 2017; Howard et al. 2017).

There are two types of image classification tasks, single-label classification and multi-label classification. In single-label classification, each input is supposed to be classified into one label. Figure 2a from ImageNet (Deng et al. 2009) shows an example input that is expected to be classified into label "tiger shark". Given an input i, the inference of a single-label classifier is a probability vector, $\mathbf{v}_i = [p_1, p_2, \ldots, p_n]$, where n is the number of labels. Each element p_j in the $\mathbf{v}_i$ represents the probability that the input belongs to the j-th label. The sum of the elements is equal to 1, i.e., $\sum_0^n p_j = 1$. The label with the highest probability is regarded as the final classification label of this classification model given this input. MNIST (LeCun and Cortes 2010), CIFAR-10 (Krizhevsky et al. 2009), and ImageNet are common datasets for single-label classification.

In multi-label classification, the number of labels of each input is not limited to one. For example, Figure 2c from COCO (Lin et al. 2014) has three labels, {"person", "motorcycle", "airplane"}. In the classification, the inference result is regarded as correct if and only if it only contains the three labels (Tian et al. 2020b; Wu and Zhu 2020). Similar to single-label classification models, given an input i, the inference of a multi-label classification model is a probability vector, $\mathbf{v}_i = [p_1, p_2, \ldots, p_n]$, where n is the number of labels. Each element p_j in the $\mathbf{v}_i$

(a) (b) (c) (d)

Fig. 2: Input Examples and their Annotations in Image Classifications. (a)(b): Image from the ImageNet Dataset and its Bounding Box, Label: "tiger shark". (c)(d): Image from the COCO Dataset and its Object Mask, Labels: "person", "motorcycle", "airplane".

represents the probability that the input belongs to the j-th label. Unlike the single-label classification model, the sum of the elements is not necessarily equal to 1, i.e., $\sum_0^n p_j \neq 1$. The final classification result is the set of labels whose probability is equal to or larger than a predefined threshold, which is usually set to 0.5 (He et al. 2016; Ben-Baruch et al. 2020). For example, given the input in Figure 2c, a multi-label classification model may output a probability vector $\mathbf{v}_i = [0.8, 0.7, 0.2, 0.6]$, where each element represents the probability of label "person", "airplane", "motorcycle" and "car", respectively. When the threshold is set to 0.5, the final classification result is {"person", "airplane", "car"}, which is an incorrect classification result as the "car" is not in the ground truth and the ground truth label "motorcycle" is not in the result. If the probability vector is $\mathbf{v}_i = [0.8, 0.7, 0.6, 0.2]$, the final result is {"person", "airplane", "motorcycle"}, and it is a correct classification result. Common multi-label datasets include COCO and Google Open Image (Krasin et al. 2017).

3 Object-Relevant Metamorphic Relations

With the aim to identify the unreliable inference made by the models based on the object-irrelevant features, we are motivated to propose two metamorphic relations as mentioned in Section 1. This section presents the details of these two relations, starting with the motivating examples. Specifically, we follow a common metamorphic testing framework to define the two metamorphic relations (Chen et al. 1998, 2018). In subsequent formulation, let $\mathcal{M}(i)$ and $\mathcal{M}(i')$ denote the inferences made by a DNN model $\mathcal{M}$ on an input image i and its follow-up input i', respectively. Let $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'))$ denote the distance between two inferences $\mathcal{M}(i)$ and $\mathcal{M}(i')$.

3.1 MR-1

Motivating Example-1 Given a source input as shown in Figure 1a, let us assume a model predicts it as "mouse". A follow-up input is constructed by corrupting the object *mouse*, as shown in Figure 1b. After feeding the follow-up input into the previous model, one of the following two cases could happen. First, it is possible that the label on follow-up input is still "mouse" and its certainty increases.

Such a situation indicates that the inference on the source input is not based on the object(*mouse*)-relevant features. If it is based on the object(*mouse*)-relevant features, it does not make sense that the model still predicts it as "mouse" when there is no such object(*mouse*). This situation is out of human expectations on image classification, as humans will not classify the follow-up image that does not have *mouse* into label "mouse" with higher certainty. Second, it is possible that the inference on the follow-up input changes to another label, or the label remains the same but the certainty decreases. In other words, due to the corruption of the object(*mouse*)-relevant features, the model cannot make the inference with the same label and the same level of certainty as the one on source input. It implies that the inference on the source input is based on the object(*mouse*)-relevant features. This situation is in line with human expectations. Since the objects have been removed or corrupted, humans are likely to classify this image to a different label, or the same label but with less certainty. Motivated by the above example, we proposed the following MR-1. In the first situation aforementioned, the MR-1 is violated while in the second situation, MR-1 is satisfied.

MR-1 An image mutated by corrupting only the features of the target object(s) should lead to an inference result with different label(s), or an inference result with the same label(s) but less certainty.

Relation Formulation of MR-1 Let i'_c be a follow-up input constructed from a source input i for a model $\mathcal{M}$ by c̲orrupting the target object but preserving its background. We consider such a mutation as *object-corrupting*. An example of object-corrupting mutation is shown in Figure 1a (source input) and Figure 1b (follow-up input). MR-1 mandates that $\mathcal{M}(i)$ and $\mathcal{M}(i'_c)$ should satisfy the relation: $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) \geq \Delta_c$. Here D takes two factors of $\mathcal{M}(i)$ and $\mathcal{M}(i'_c)$ into consideration, i.e., the labels in the inferences and the certainty of the inferences. The detailed definition of D for MR-1 is introduced in Section 4.4.1. Δ_c denotes a threshold for the distance between two inference results made by a model under metamorphic testing using object-corrupting mutations.

Explanation of MR-1 If an inference made by a specific model is based on object-relevant features, after object-corrupting mutations, the new inference results should be affected since those object-relevant features have been corrupted, and thus those features cannot be further utilized by the model anymore. Such effects could cause two consequences. First, the model can still make the same inference as the inference of the original input while the certainty of the inference given by the model should be decreased since the object-relevant features have been corrupted. Second, the model cannot make the same inference as the inference of the original input if the corruption is very severe. Consequently, the label of the new inference should be different from the original one.

3.2 MR-2

Motivating Example-2 Given a source input shown in Figure 1a, assume a model predicts it as "mouse". A follow-up input is constructed by preserving the object, as shown in Figure 1c. After feeding the follow-up input into the previous model, one of the following two cases could happen. First, the inference on follow-up input is not "mouse" anymore. It indicates that the inference on the source input is not based on the object(*mouse*)-relevant features. Since the object *mouse* is still in the input, if the inference on the source input is based on the object(*mouse*)-relevant features, the inference should still be the "mouse". Second, the inference on follow-up input remains the same label. It implies that the inference on the source input is based on the object(*mouse*)-relevant features. When the object-relevant features are preserved, the model can leverage them to make the correct inference. Such a situation is in line with human expectations. Motivated by this example, we propose the following MR-2. In the above example, MR-2 is violated in the first situation and satisfied in the second situation.

MR-2 An image mutated by preserving the features of the target object(s) and corrupting other features should lead to an inference result with the same label(s).

Relation Formulation of MR-2 Let i'_p be a follow-up input constructed from a source input i for a model $\mathcal{M}$ by preserving the target object(s) but mutating the other parts. We consider such a mutation *object-preserving*. An example of object-preserving mutation is shown in Figure 1a (source input) and Figure 1c (follow-up input). MR-2 mandates that $\mathcal{M}(i)$ and $\mathcal{M}(i'_p)$ should satisfy the relation: $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) \leq \Delta_p$. Here, Δ_p denotes a threshold for the distance between two inference results made by a model under metamorphic testing using object-preserving mutations. The detailed definition of D for MR-2 is introduced in Section 4.4.2.

Explanation of MR-2 If an inference made by a specific model is based on object-relevant features, after object-preserving mutations, the labels of the new inference result should not be changed, since the object-relevant features are preserved and the model should be able to use them.

4 Approach

We present our approach in this section, starting from an overview of the whole approach, followed by the explanation of each stage.

4.1 Overview

Figure 3 shows the overview of our approach, including the following three stages:

1 **Object-Relevant Feature Identification** Given an inference to be examined, we regard its input image as the source input. We semantically divide the input into two parts, a *target-object region* and a *background region*. The *target-object region* is where the target object(s) is located and where the object-relevant features are encoded. The *background region* is where the object-irrelevant features are encoded.

2 **Follow-up Inputs Construction** Mutation functions are leveraged to generate follow-up inputs from the source inputs, based on the proposed metamorphic relations. Specifically, these mutation functions will corrupt, or preserve the object-relevant features in the source input. The corresponding testing oracles will also be generated based on the metamorphic relations.

3 **Metamorphic Relation Validation** We validate if the distance between the inference result of a source input and the inferences of its follow-up inputs violates the test oracles. If so, the inference of the source input is flagged as an

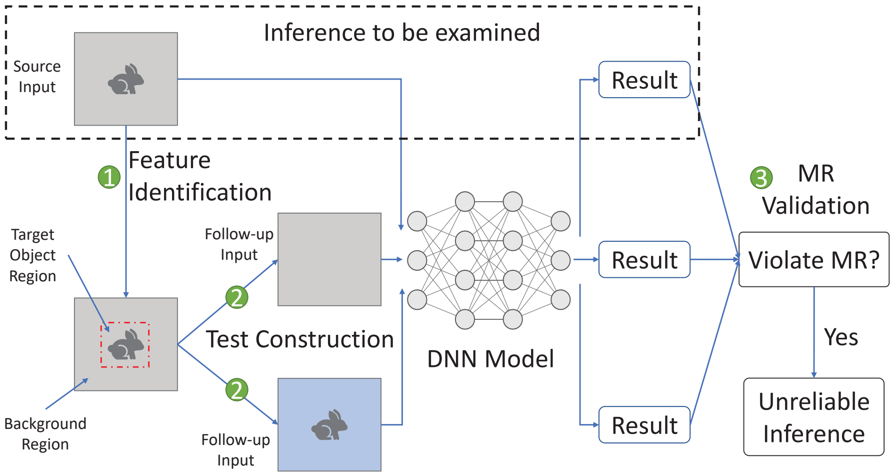


Fig. 3: The Overview of our Metamorphic Testing Approach

unreliable inference, which means this inference is made mainly based on object-irrelevant features.

Please note that our approach mainly assesses the correct inference results from image classification models. In single-label classification, "correct" means that the top-1 label in the result is the same as the source input's ground truth. In multi-label classification, "correct" means that the set of labels in the results is the same as the set of labels in the source input's ground truth, as we mentioned in Section 2.2. We focus on correct inferences since if the inference result is incorrect, the target object might not exist in the input, and thus it is challenging to identify the object-relevant features.

4.2 Object-Relevant Feature Identification

In single-label classification, since each image only has a single label, we regard the object(s) belonging to the annotated label as the target object(s). For multi-label classification, each image can have multiple labels. We regard the union of all objects belonging to the annotated labels as the target objects. For example, for the input as shown in Figure 2c, the target objects consist of the airplane, motorcycle and person. In both cases, the pixels where the target object(s) reside are treated as the *target-object region* and the others are regarded as the *background region*.

The annotations of the target objects could be extracted from the dataset, or obtained using the latest object localization techniques, such as YOLO (Redmon et al. 2016) and Faster R-CNN (Ren et al. 2017). Currently, several datasets for image classification provide the annotation of objects, such as ImageNet, COCO, PASCAL VOC and Google Open Image. The annotations are usually in the format of a *bounding box*. For example, the bounding box of the *tiger shark* in Figure 2a is displayed as the red rectangle in Figure 2b. Some datasets, such as COCO, annotate the object using the object mask, which draws the boundary of each object with a finer granularity. These annotations provide the exact target-object region that does not contain any pixels belonging to the background region. Figure 2d shows the object marks of "person", "motorcycle" and "airplane".

Both annotation formats can be used in our approach. If the annotations are provided as bounding boxes, we regard the region of the bounding boxes as the target-object region. Although the target-object region could contain some pixels that do not belong to the target object(s), the majority of the region represents the target object. If the annotations are object masks, we regard the region covered by the object masks as the target-object region. In our experiment, we used the bounding box for the experiments based on the ImageNet dataset and the object mask on the COCO dataset, depending on the availability of the annotation format in these datasets.

4.3 Follow-up Inputs Construction

We generate the follow-up inputs by semantically corrupting or preserving the object-relevant features of a source image using the two aforementioned image mutations: *object-corrupting* mutation and *object-preserving* mutation.

There are many possible ways to design the mutation functions to corrupt or preserve the object-relevant features. However, it is challenging to quantitatively measure the degree of corruption and preservation. Such a challenge further brings difficulties to define the test oracle, as different levels of corruption and preservation should correspond to different designs of test oracle, especially the thresholds of test oracle (e.g., the Δ_c in Section 3). An inappropriate test oracle will influence the effectiveness of our approach.

To alleviate this challenge, we mutate the image by filling simple colors, such as white, gray and black, into the target-object region (or background region). Correspondingly, we use whether the classification results of source input and follow-up input are equal as the test oracle. The objective of our mutation is to simulate extreme cases, without considering the realism of images. For example, if the target-object region in the source input is substituted by black color, i.e., the object-relevant features are removed, but the model can still classify it correctly, the model is very likely to make the inference based on the object-irrelevant features. In real scenarios, our mutation can be considered as the simulation of the blocking of cameras. An existing study (Pei et al. 2017) designed for testing DNNs also generates test images via randomly patching black holes to images, in order to simulate the blocking of cameras.

Besides alleviating the above challenge, another advantage of using simple colors is that these colors bring little additional features to the source input. If we replace the object region with other objects or patterns, they may bring new features and further affect the model inference results. In such a situation, one cannot easily identify whether the change of the inference result is due to the absence of object-relevant features, or the appearance of these new features.

In our experiments, we use three colors, i.e., black (R0, G0, B0), gray (R127, G127, B127) and white (R255, G255, B255). For each source input, three follow-up inputs are generated based on MR-1 and three more are generated based on

Fig. 4: (a): Original Image with Bounding Box, Label: "pitcher, ewer". (b): Image after Object Corrupting Mutation for MR-1. (c): Image after Object Preserving Mutation for MR-2. (d): Image Inpainting Result using DeepFill.

MR-2. For example, given the source input shown in Figure 4a, Figure 4b and Figure 4c are two follow-up inputs generated based MR-1 and MR-2, respectively. It is possible that such simple colors could also induce bias to model inference. To alleviate this threat, eventually, we use the majority of their validation results as the final result. Such a strategy is called the *majority voting* (Freund and Schapire 1995) and it has been used by an existing study (Pei et al. 2017) to test DNN systems. One threat to validity that might be raised is whether three colors are sufficient for performing metamorphic testing. To alleviate this threat, we compare the results using more colors in Section 5.2, and demonstrate that using three colors is sufficient.

Another threat that might be raised is why not using the inpainting technology to remove the object/background more naturally. Actually, we tried this method at the exploratory stage of this study. However, even the-state-of-art technology DeepFill (Yu et al. 2018) cannot completely remove the object features. An example is shown in Figure 4d. The feature of pitcher in the image cannot be removed completely. Moreover, such inpainting models usually need hundreds of hours for training and ~15 seconds to inpaint an image, which is not efficient.

4.4 Metamorphic Relation Validation

In this subsection, we introduce the metamorphic relation validation process. Please note that in our experiments on single-label and multi-label classification models, for each source input i, we generate three follow-up inputs i's. Then we will validate the MRs three times and use majority voting to decide whether MRs are violated. As we mentioned, such a method can mitigate the possible threat induced by a single mutation. We will regard $\mathcal{M}(i)$ as an unreliable inference if and only if MR-1 is violated at least two out of three times. The same strategy is applied for MR-2.

4.4.1 Validation of MR-1

MR-1 An image mutated by corrupting only the features of the target object(s) should lead to an inference result with different label(s), or an inference result with the same label(s) but less certainty.

Single-label Classification Here we use the same notation as Section 3. We define the distance function $\mathcal{D}$ as follows:

$$\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) = \begin{cases} 1, & \text{if } l_{\mathcal{M}(i)} \neq l_{\mathcal{M}(i'_c)} \\ & \text{or if } l_{\mathcal{M}(i)} = l_{\mathcal{M}(i'_c)} \text{ and } \mathcal{C}(l_{\mathcal{M}(i)}) > \mathcal{C}(l_{\mathcal{M}(i'_c)}) \\ 0, & \text{otherwise} \end{cases}$$

Here, $l_{\mathcal{M}(i)}$ is the label of the target object in $\mathcal{M}(i)$. $\mathcal{C}(\mathcal{M}(i))$ measures the certainty of the $M(i)$, according to the definition proposed by existing work in DNN testing (Xie et al. 2019b; Zhang et al. 2020b):[1]

$$\mathcal{C}(\mathcal{M}(i)) = \min_{0<j<n, j\neq l} \left|p_l - p_j\right|$$

where p_l is the probability of label $l_{\mathcal{M}(i)}$ and p_j is the probability of j-th label in the inference. Intuitively, the certainty measures the minimal difference between label $l_{\mathcal{M}(i)}$ and any other labels in terms of their probabilities. The value of $\mathcal{C}(\mathcal{M}(i))$ ranges in the region $[0, 1]$. The higher the value is, the more certain the model is on the inference. If the inference is a correct inference, the above certainty equation actually calculates the difference between the highest probability and the second highest probability.

Correspondingly, we define Δ_c equals to 1. if $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) \geq \Delta_c = 1$, i.e., the label of the inference on the source input $l_{\mathcal{M}(i)}$ is different from the one of the inference on the follow-up input $l_{\mathcal{M}(i'_c)}$, or the labels are the same but the inference on the follow-up input become less certain, the MR-1 is satisfied. Otherwise, if $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) < \Delta_c = 1$, i.e., $l_{\mathcal{M}(i)}$ and $l_{\mathcal{M}(i'_c)}$ are the same and the certainty increases, it implies that after corrupting the object-relevant features in the source input, the model can still correctly classify the input with more certainty. In other words, the examined inference $\mathcal{M}(i)$ is made based on features irrelevant to the objects. This conclusion violates our MR-1, and thus $\mathcal{M}(i)$ is labeled as an unreliable inference.

Multi-label Classification In multi-label classification, we adapt the above formula with slight modifications to cooperate with the multiple labels. Specifically, we use $L_{\mathcal{M}(i)}$ to denote the set of labels outputted by the model $\mathcal{M}$ on input i. We define the distance function $\mathcal{D}$ as follows:

$$\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) = \begin{cases} 1, & \text{if } L_{\mathcal{M}(i)} \neq L_{\mathcal{M}(i'_c)} \\ & \text{or if } L_{\mathcal{M}(i)} = L_{\mathcal{M}(i'_c)} \text{ and } \mathcal{C}(L_{\mathcal{M}(i)}) > \mathcal{C}(L_{\mathcal{M}(i'_c)}) \\ 0, & \text{otherwise} \end{cases}$$

To the best of our knowledge, the certainty in multi-label classification has not been defined by existing work, and the definition in single-label classification cannot be applied to multi-label classification directly. As we introduced in Section 2, in single-label classification, the sum of the probability of all labels is equal to 1. Labels are competing with each other and only the label with the highest probability is regarded as the final result. In other words, the increase of the probability of a label means the decrease of the probability of other labels. Thus, we can measure the certainty based on to what extent the probability of this label is different from the probabilities of the remaining labels. However, as we mentioned

[1] The latter study refers this concept as "prediction confidence"

in Section 2.2, in multi-label classification, the probabilities of labels are relatively independent, i.e., the sum of the probabilities of all labels are not necessarily equal to 1. The difference between the probabilities of the two labels does not imply the inference certainty.

To address this challenge, in our approach, we regard the multi-label classification into multiple binary-classification tasks where each binary-classification predicts whether the input belongs to a single label or not. This enables us to measure the certainty of each label individually. For example, let us assume an inference result given by a multi-label classification is $[0.8, 0.9, 0.2]$, which corresponds to the probability of "airplane", "person" and "motorcycle". We can regard it as the outputs from three binary-classification models. The first model predicts whether the input belongs to label "airplane" and outputs the probability 0.8. The second and third ones predict whether the input belongs to label "person" and "motorcycle", and output the probability 0.9 and 0.2, respectively. It is trivial to calculate the certainty of the binary classification task. Therefore, we can first measure the certainty of each binary classification, and then leverage the results to measure the certainty of multi-label classification.

More specifically, for any label l in the inference result of $\mathcal{M}(i)$ and its probability p, we define the certainty $\mathcal{C}_{l,\mathcal{M}(i)}$:

$$\mathcal{C}_{l,\mathcal{M}(i)} = |p - (1-p)| = |2p - 1|$$

The value of $\mathcal{C}_{l,\mathcal{M}(i)}$ is within the region $[0, 1]$. The intuition is to measure the certainty based on the difference between the probability that "it belongs to label l" and "it does not belong to label l". The larger the difference is, more certain the model is on the inference. Based on the above definition of certainty of single label in the multiple-classification, we define the comparison of $\mathcal{C}(L_{\mathcal{M}(i)})$ and $\mathcal{C}(L_{\mathcal{M}(i'_c)})$ as following: $\mathcal{C}(L_{\mathcal{M}(i)}) > \mathcal{C}(L_{\mathcal{M}(i'_c)}) \iff \mathcal{C}_{l,\mathcal{M}(i)} > \mathcal{C}_{l,\mathcal{M}(i'_c)}, \forall\ l \in\ L_{\mathcal{M}(i)}$. The above equation compares the certainty of each label in the inferences on the source input and the follow-up input. Please note that for the predicate of certainty $\mathcal{C}(L_{\mathcal{M}(i)})$ and $\mathcal{C}(L_{\mathcal{M}(i'_c)})$, we check it only if the prior predicate $L_{\mathcal{M}(i)} = L_{\mathcal{M}(i'_c)}$ is true.

For Δ_c, we use the same definition as single-label classification, i.e., $\Delta_c = 1$. If $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_c)) \geq \Delta_c$, the MR-1 is satisfied. Otherwise, MR-1 is violated and the examined inference, i.e., $\mathcal{M}(i)$, is regarded as an unreliable inference.

4.4.2 Validation of MR-2

MR-2 An image mutated by preserving the features of the target object(s) and corrupting other features should lead to an inference result with the same label(s).

Single-label Classification We define $\mathcal{D}$ as follows:

$$\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) = \begin{cases} 0, & \text{if } l_{\mathcal{M}(i)} = l_{\mathcal{M}(i'_p)} \\ 1, & \text{otherwise} \end{cases}$$

Here, $l_{\mathcal{M}(i)}$ is the label with the highest probability in $\mathcal{M}(i)$. We define the threshold $\Delta_p = 0$. If $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) > \Delta_p = 0$, it means that the label of the inference on the source input $l_{\mathcal{M}(i)}$ is different from the one of the inference on the follow-up input $l_{\mathcal{M}(i'_p)}$. In other words, after preserving the features of the target object and corrupting the remaining features in the source input, the model classifies the follow-up input into a different label. This conclusion is opposite to our MR-2, and thus the examined inference $\mathcal{M}(i)$ is labeled as an unreliable inference. If $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) \leq \Delta_p = 0$, it implies that after preserving the features of the target object and corrupting the others, the model still classifies the input into the same label as the one of the source input. This result is in line with our MR-2 and thus the examined inference will not be labeled as an unreliable inference by us.

Multi-label Classification We define $\mathcal{D}$ as follows:

$$\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) = \begin{cases} 0, & \text{if } L_{\mathcal{M}(i)} = L_{\mathcal{M}(i'_p)} \\ 1, & \text{otherwise} \end{cases}$$

Here, $L_{\mathcal{M}(i)}$ is the set of labels in $\mathcal{M}(i)$. The equality of the $L_{\mathcal{M}(i)}$ and $L_{\mathcal{M}(i'_p)}$ is based on the equality of set. In other words, $L_{\mathcal{M}(i)} = L_{\mathcal{M}(i'_p)}$ if and only if for any element in $L_{\mathcal{M}(i)}$, this element is also in $L_{\mathcal{M}(i'_p)}$ and for any element in $L_{\mathcal{M}(i'_p)}$, it is also in $L_{\mathcal{M}(i)}$.

Same as single-label classification, the Δ_p is defined as 0. If $\mathcal{D}(\mathcal{M}(i),\ \mathcal{M}(i'_p)) > \Delta_p = 0$, it means $L_{\mathcal{M}(i)}$ and $L_{\mathcal{M}(i'_p)}$ are different. In other words, after preserving the features of the target object and corrupting the remaining features in the source input, the model classifies the input into different labels with the inference on the source input. This conclusion is opposite to our MR-2, and thus the examined inference $\mathcal{M}(i)$ is labeled as an unreliable inference.

5 Evaluation

In this section, we evaluate our approach from the perspective of effectiveness. First, we investigate the effectiveness of our proposed approach to see whether it can successfully identify inferences that are made based on object-irrelevant features. Specifically, we measure the precision (true positive rate) of our approach, i.e., the number of real unreliable inferences in all inferences identified by our approach. We aim to answer the following question:

RQ1 What is the effectiveness of our approach in terms of true positive rate?

Further, as mentioned in Section 4, we use three colors to mutate inputs in our approach. One threat of our approach is that whether more colors should be used to identify unreliable inferences. To answer this question, we performed another experiment in which we use 15 distinct colors to mutate the inputs, and then compared it with the experiment in which only 3 colors are used. These results will help us to answer the following question:

RQ2 Is it sufficient to use only 3 colors for mutations in terms of effectiveness?

The source code and data of our experiment are available online.[2] Our experiments were conducted on two datasets, the ImageNet 2012 validation set and COCO 2014 validation set. The ImageNet 2012 validation set is a popular single-label classification dataset with 50,000 images. These images evenly distribute across 1,000 labels. The COCO 2014 validation set is a common multi-label classification dataset, with 40,504 images across 80 labels. On average, each image has 7.21 labels. We chose these datasets for three reasons. First, both are popular

[2] https://github.com/yqtianust/PaperUnreliableInference

image classification datasets on which most state-of-the-art models are trained. Second, there are plenty of pre-trained models available as experiment subjects. Third, they provide the annotation of object boundaries.

5.1 Effectiveness of Our Approach

In order to evaluate whether our metamorphic testing approach can effectively identify unreliable inferences, we applied it to the inferences made by three pre-trained single-label classification models from the Keras Application (Chollet et al. 2015a) and one multi-label classification model (Ben-Baruch et al. 2020). The former models are trained on the ImageNet dataset and the latter one is trained on the COCO dataset. Then we manually validated the testing results and measured the precision.

To validate whether the unreliable inferences identified by our approach are indeed made based on object-irrelevant features, for each of them, we manually checked the quality of their follow-up inputs. If the follow-up inputs are constructed as expected, i.e., the object features in the follow-up inputs are corrupted(preserved) for MR-1(MR-2), we regarded the corresponding inference as indeed unreliable, i.e., a true positive case. If the follow-up inputs are not constructed as expected, the corresponding inference cannot be regarded as an unreliable inference, thus resulting in a false positive case.

More specifically, for the inference results that violate MR-1, we manually checked whether the object-relevant features were completely corrupted after the mutation, i.e., whether the target objects in the follow-up inputs are indeed removed. If the follow-up input does not contain the target object, the inference violates MR-1 since the model still predicts it as the original label. Thus, this test result is a true positive. If the follow-up input still contains the target object, predicting it as the original label does not violate MR-1, and hence the identified unreliable inference is a false positive.

Similarly, for the inference that violated the relation MR-2, we manually checked whether the target objects were preserved and whether the other features were corrupted. Specifically, if the follow-up input contains the target object, the MR-2 is violated since the model does not predict the follow-up input as the original label. So, we labeled the test result as true positive. On the contrary, if the follow-up input does not contain the target object, MR-2 is not violated and the identified unreliable inference is a false positive.

The manual check was conducted by two graduate students individually and independently. Only the results agreed by consensus were considered. The disagreed results were labeled as “uncertain”.

5.1.1 Pilot Study

Before the manual check, we first conducted a pilot study to help us understand the possible cases (i.e., the root cause of false positive cases) that might be encountered in the manual check. Specifically, we randomly selected 200 unreliable inferences found by our approach to perform the pilot study, among which 100 violate MR-1 and the others violate MR-2. We investigated whether each unreliable inference is true positive and if not, what are the major reasons for those false positive cases.

In the investigation, for each unreliable inference, each student was requested to view a pair of inputs (pictures in our scenarios). More specifically, each pair of the inputs consisted of two inputs: (a) the source input on which the unreliable inference is made, e.g., Figure 4a, and (b) the follow-up input constructed based on the source input, e.g., Figure 4b if MR-1 is violated, or Figure 4c if MR-2 is violated. Besides, the label of the source input was provided to the students. The students were required to answer the following questions for the unreliable inferences violating MR-1:

1. Do you think the object-relevant features of the source input have been completely corrupted in the follow-up inputs, i.e., the target objects in the follow-up inputs have been indeed removed?
2. If not, please briefly explain the reason.

Similarly, for the unreliable inferences violating MR-2, the corresponding questions were:

1. Do you think the object-relevant features of the source input have been completely preserved in the follow-up inputs, i.e., the target objects still remain in the follow-up inputs?
2. If not, please briefly explain the reason.

First, two graduate students investigated the selected 200 image pairs individually and independently. Their answers to the questions have been recorded. Then for the inconsistent answers, they discussed with each other to see if they can reach a consensus. A reason is selected as a common reason if it occurs more than or equal to 10 times. Eventually, we finalized three common reasons inducing false positives for unreliable inference violating MR-1, which are:

(a)*Existence of Multiple Target Objects.* These false positives occurred because there are multiple target objects in the source input, but not all of them are corrupted in the follow-up inputs. Figure 5f shows an example. The original image in Figure 5b, whose label is "confectionery", has multiple confectioneries. Ideally, all of them should be corrupted in its follow-up inputs. However, after mutation, the follow-up input, as shown in Figure 5f, still contains multiple confectioneries since the dataset only annotates one of them, which is shown as the red rectangle in Figure 5b. As such, the inference of the follow-up input can still be "confectionery" as the object-relevant features (the other confectioneries) are not completely corrupted. Therefore, MR-1 is not violated and the original inference is a false positive of the identified unreliable inferences.

(b)*Incomplete Removal of the Target Object.* Some false positives occurred in the inputs that contain a single target object but only parts of it are corrupted in the follow-up input. An example is shown in Figure 5c, whose label is "drilling platform". Ideally, the entire platform should be corrupted in the follow-up inputs. However, the mutated images shown in Figure 5g still contain part of the target object. This is because the annotation provided by the ImageNet dataset does not cover the upper-half of "drilling platform", which differs from the other images in this label whose platforms are entirely annotated. Therefore, the follow-up input can lead to the same classification result as the original inference because the object-relevant features are not corrupted entirely. The MR-1 is not violated in this case.

(c) *Others*. It refers to the other reasons not belonging to the above two reasons. For example, the original image is not clear and hinders the students to identify the boundary of the target object.

For MR-2, we do not distinguish the reason for false positives since the number of false positives is very limited (less than 10 in our pilot study).

We also conducted a similar pilot study for multi-label classification. More specifically, we selected 50 unreliable inferences violating MR-1 and 50 ones violating MR-2 from all the unreliable inferences in multi-label classification found by our approach. A reason is considered common if it occurs at least 5 times. Since we did not notice other reasons than the ones aforementioned, we concluded the same reasons for both single-label and multi-label classifications.

5.1.2 Experiment Setup

Model Selection For the single-label classification model, we selected NASNet-Large (Zoph et al. 2018), MobileNet (Howard et al. 2017) and ResNet101 (He et al. 2016) among the pre-trained models from the Keras Application (Chollet et al. 2015b) because their top-1 accuracy lies at the top, medium and bottom, respectively, among those of the models. For the multi-label classification model, we selected TResNet-XL (Ben-Baruch et al. 2020), since it achieves the highest accuracy on the COCO dataset to the best of our knowledge (Ben-Baruch et al. 2020) till March 2021.

Sampling We randomly sampled the inferences made by the four models for the manual check, where the sample size is determined by the Cochran formula (Cochran 1963) with 95% confidence level.

Manual Check Two graduate students conducted the manual check similar to the pilot study. More specifically, each source input in the unreliable inference was displayed with the follow-up inputs constructed by our method. The students were asked the same question as the ones in the pilot study. The only difference is that at this time, the Q2 in unreliable inference violating MR-1 was supplied with three options, which are: (a) *Existence of Multiple Target Objects*, (b) *Incomplete Removal of The Target Object*, (c) *Others*. When (c) is chosen, the students were also required to write down detailed explanations. The students were allowed to choose multiple of the above options. During the manual check, we also monitored the reasons in (c) *Others*. If any reason in (c) *Others* occurs at least 10 times, we would extract a new common reason. Such a situation does not exist in our manual check.

Each student conducted the manual check individually and independently. It took around 15 hours for each of them to complete the manual check. After the individual check, they discussed the cases where the disagreement arises, in case any of them miss anything during the check. If the disagreement is addressed, the corresponding manual check result is changed. At last, we collected and analyzed the results. As we mentioned previously, only the results agreed by consensus were considered in the analysis. The Kappa Agreement Score (Landis and Koch 1977) of the manual check is 0.955. Such a value indicates an almost perfect agreement between the two graduate students who conducted the manual check.

Threat to validity There is a potential threat to validity in this experiment. Our manual check is subject to human mistakes. To address the threat, two graduate students conducted the manual check individually and independently. A result

(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 5: (a)(b)(c)(d): Images (with Bounding Boxes) as the Source Inputs. (e)(f)(g)(h): Images as the Corresponding Follow-up Inputs. Labels: (a): "goldfinch, Carduelis carduelis", (b): "confectionery", (c): "drilling platform", (d): "car wheel".

Table 1: The Manual Check Results for the Effectiveness of MR-1 on Single-label Classification Models. Column *Multiple* is for the Reason *Existence of Multiple Target Objects* and Column *Incomplete* is for the Reason *Incomplete Removal of the Target Object*. The Number in the Parentheses under *Multiple* is for Cases Shared by Both Reasons.

Model	Accuracy	Total	Sample Size	True Positive	False Positive			Uncertain
					Multiple	Incomplete	Others	
NASNetLarge	82.7%	826	311	202 (65.0%)	84 (1)	16	4	6
ResNet101	76.4%	344	194	122 (62.9%)	55 (2)	8	8	3
MobileNet	70.3%	222	149	95 (63.8%)	42 (1)	8	3	2
Total		1,392	654	419 (64.1%)	181 (27.7%)	32 (4.9%)	15	11

will be adopted only if both students made the same conclusion. The high Kappa Agreement Score indicates that the results is reliable.

5.1.3 Results and Discussion

Single-label Classification Models Table 1 and Table 2 show the manual check results for MR-1 and MR-2 for single-label classification models, respectively. The column *Total* refers to the number of unreliable inferences identified by our approach for each model. Specifically, our approach identifies 1,392 inferences that violate MR-1 and 15,198 inferences that violate MR-2. We randomly sampled and manually checked 654 and 1,069 inferences from these two categories, respectively, as previously explained.

Table 2: The Manual Check Results for the Effectiveness of MR-2 on Single-label Classification Models.

Model	Accuracy	Total	Sample Size	True Positive	False Positive	Uncertain
NASNetLarge	82.7%	3,634	348	339 (97.4%)	1	8
ResNet101	76.4%	4,942	357	340 (95.2%)	7	10
MobileNet	70.3%	6,622	364	351 (96.4%)	0	16
Total		15,198	1,069	1,030 (96.4%)	8(0.07%)	34

Table 3: The Manual Check Results for the Effectiveness of MR-1 and MR-2 on Multi-label Classification Model: TResNet-XL).

MR	Total	Sample Size	True Positive	False Positive	Uncertain
MR-1	957	275	215 (78.2%)	44	16
MR-2	4,732	356	308 (86.5%)	30	18

As for the inputs that violate MR-1, the column *True Positive* of Table 1 shows that our approach achieves an average precision of 64.1%, ranging from 62.9% to 65.0% for different models. Out of the 654 samples, 419 samples do not contain the target objects in the follow-up inputs but the models keep labeling them as the target objects. So, they violate MR-1 and are true positive cases. Figure 5a shows an example, in which the original image is correctly classified by the model ResNet101 as "goldfinch, Carduelis carduelis". Although the follow-up input in Figure 5e does not contain birds, ResNet101 gives the same classification result as that of the original image, thus resulting in an unreliable inference.

We further checked the remaining 235 (=654 - 419) false positive cases, and found that 77.0% (=181/235) of the false positive cases are due to the *Existence of Multiple Target Objects* and 13.6%(=32/235) are because of *Incomplete Removal of the Target Object*. Moreover, there are four cases that belong to both *Existence of Multiple Target Objects* and *Incomplete Removal of the Target Object*. The above numbers (181 and 32) have included these four cases. Besides, there are 11 cases labeled as uncertain as the results from two students disagree with each other. The rest of the false positive cases (15 in total) are labeled as Others.

As for the inputs that violate MR-2, it shows that our approach achieves an aggregated precision of 96.4%, ranging from 95.2% to 97.4% for different models. In total, 1,030 out of the 1,069 samples preserve the target objects in the follow-up inputs, but these follow-up inputs are not correctly classified by the models. Therefore, these samples indeed violate MR-2 and they are regarded as true positives of the unreliable inferences violating MR-2. For the remaining 39 cases, only part of the target objects is preserved in the follow-up inputs. They do not violate MR-2 and are false positives. For instance, given the source input as shown in Figure 5d, the constructed follow-up input in Figure 5h only covers the center of wheel but not the entire tire. According to the definition from the WordNet (Fellbaum 2006) (the labels of the ImageNet dataset are defined according to WordNet), "car wheel" is *"a wheel that has a tire and rim and hubcap"*. Since the object-relevant features are only partially preserved, it makes sense that the follow-up input is incorrectly classified. Therefore, MR-2 is not violated and this is a false positive case.

We noticed that the precision of MR-2 is much higher than that of MR-1. We found the reason is that the aforementioned *Existence of Multiple Target Objects* will cause the follow-up input unqualified for the validation of MR-1, as the object-relevant features of the follow-up inputs will not be completely corrupted. However, such a situation will not affect MR-2 since as long as one of the target objects is preserved in the follow-up inputs, the follow-up inputs are valid to validate MR-2.

Multi-label Classification Models Table 3 shows the manual check results for MR-1 and MR-2 for TResNet-XL, a multi-label classification model, respectively. The true positive rate for MR-1 and MR-2 is 78.2% and 86.5% respectively. This shows that our approach is also effective for multi-label classification models. As for the false positives for MR-1, the major reasons are still *Existence of Multiple Target Objects* and *Incomplete Removal of The Target Object*. They account for 20 and 23 of the 44 false positive cases. The remaining one is due to the incorrect annotation, where a labeled broccoli is actually lettuce. For the false positives for MR-2, similar to single-label classification, the major reason is that their target objects are not completely preserved in the follow-up inputs and thus they do not violate MR-2.

Answer to RQ1 Our approach is effective in identifying unreliable inferences that violate MR-1 and MR-2, with an aggregated precision of at least 62.9% and 86.5%, respectively. The false positives are mainly caused by imperfect annotation of the target objects.

5.2 The Impact of The Number of Colors in Our Approach

As mentioned in Section 4, we use three colors to mutate inputs in our approach and use the majority of their results to identify the unreliable inference. One threat of our approach is that whether three colors are sufficient to identify unreliable inferences. To answer this question, we performed another experiment that uses 15 distinct colors to mutate the inputs, and we then compared the results obtained of the new experiment with that of the original one.

5.2.1 Experiment Design

Specifically, besides the three colors we used previously, we select 12 more commonly used colors, which are red (R255, G0, B0), maroon (R128, G0, B0), yellow (R255, G255, B0), olive (R128, G128, B0), lime (R0, G255, B0), green (R0, G128, B0), aqua (R0, G255, B255), teal (R0, G128, B128), blue (R0, G0, B255), navy (R0, G0, B128), fuchsia (R255, G0, B255), and purple (R128, G0, B128). We use the same approach as mentioned in Section 4. The only difference is that now we regard an inference as unreliable if and only if the MR is violated by at least 8 out of the 15 mutated inputs.

After the data collection, we compared the results using 15 colors and the ones using 3 colors. Statistically, we use the Chi-square independence test (F.R.S. 1900) to test the independence of the results obtained from the two approaches. The Chi-square independence test is commonly used to determine if there is a

Table 4: Contingency Tables for MR-1 and MR-2 to Compare the Experiment Results Obtained using 3 Colors vs 15 Colors.

	MR-1		MR-2	
	V_3: Violate	V_3: Not Violate	V_3: Violate	V_3: Not Violate
V_{15}: Violate	169	52	5,145	1,467
V_{15}: Not Violate	63	35,323	249	28,773

significant relationship between two categorical variables. In our experiment, we use it to determine if the decision "violate MR or not" using by three colors and the one using fifteen colors are strongly correlated. If yes, we can use three colors to save computation resources. We conduct the experiment using the pre-trained VGG16 from Keras.

5.2.2 Results and Discussion

We use variable V_3 to denote the decision "violate MR or not" according to the approach using three colors. Similarly, we use variable V_{15} to denote the decision "violate MR or not" according to the approach using 15 colors. We build the contingency tables for both MR-1 and MR-2 as shown in Table 4. The cell in the table represents the number of the inferences identified by the two approaches. For example, the cell "169" means there are 169 inferences that are considered as violating MR-1 by both the approach using three colors and the one using fifteen colors. The cell "1,467" means there are 1,467 inferences that are considered as not violating MR-2 by the approach using three colors and considered as violating MR-2 by the approach using the fifteen colors.

The p-values of the Chi-square test are both < 0.001 for MR-1 and MR-2, which is less than the typical threshold 0.05. The corresponding effect sizes[3] are 0.743 and 0.835 for MR-1 and MR-2, respectively. It indicates that the results obtained by the approach using three colors and the approach using fifteen colors are strongly correlated. In other words, if an inference is considered unreliable (or reliable) by the approach using three colors, the same decision will likely be made by the approach using fifteen colors, and vice versa. Overall, this experiment shows that using more colors than three in our approach has a minor difference compared to three colors. Therefore, it is sufficient to use three colors for the follow-up input construction in our approach.

Answer to RQ2 Using three colors in our approach is sufficient to identify unreliable inputs effectively.

6 Empirical Study

Leveraging our approach, we conduct an empirical study to understand the unreliable inference problems in reality.

First, we want to understand the pervasiveness of the problem, i.e., to what extent are the inference results made by the state-of-the-art DNN models based

[3] in the Chi-square test, it is usually referred to as Cramér's V (Cramer 1946)

on object-irrelevant features. Specifically, we measure the proportion of unreliable inferences identified in all correct inferences outputted by these models.

RQ3 How pervasive is unreliable inference in DNN models?

Second, we study the characteristics of the identified unreliable inferences. Specifically, we focus on the size of the target objects in unreliable inferences, a common attribute of objects. We studied whether there is any correlation between the object size and the unreliable inferences.

RQ4 Is there a correlation between the target object size and the unreliable inferences?

Next, we aim to understand the effect of such unreliable inferences. Specifically, we investigate whether the unreliable inferences can significantly affect a model's evaluation result, thus preventing us from correctly evaluating models and comparing them fairly. In the experiments, we compare the accuracy of a model before and after removing those unreliable inferences from the associated test.

RQ5 To what extent will the unreliable inference affect a model's evaluation?

Finally, we investigate how to tame unreliable inferences. Specifically, we investigate whether the ratio of unreliable inferences can be reduced during the training process and whether it is correlated with the evaluation metrics such as accuracy. To achieve this goal, in the experiments, we track the ratio of unreliable inferences and the model accuracy during the model training process .

RQ6 Can the unreliable inference be tamed during training?

6.1 Pervasiveness of Unreliable Inferences

RQ3 How pervasive is unreliable inference in DNN models?

6.1.1 Motivation

In the previous section, we showed that thousands of inferences made by the four pre-trained classification models violate our MRs. In this subsection, we investigate the pervasiveness of the problem, i.e., whether such unreliable inferences generally exist in a wide variety of models with different architectures. We leveraged our methodology to identify the unreliable inferences made by both the single-label and multi-label image classification models. Then we measure the ratio of the unreliable inferences in all correct inferences. This research question can help us to understand the severity of the unreliable inferences.

6.1.2 Experiment Setup

We collected 21 pre-trained DNN models from public repositories. 18 out of the 21 models are single-label image classification models, and they are collected from the Keras Application (Chollet et al. 2015b), a famous and popular repository for pretrained models. All of them are trained on the ImageNet dataset, and their information (name and accuracy) is shown in the first two columns of Table 5. Besides the single-label classification models, we also collected three multi-label classification models, which are ResNet-50 (He et al. 2016), TResNet-L (Ben-Baruch et al. 2020) and TResNet-XL (Ben-Baruch et al. 2020). ResNet-50 is chosen as it has been used as an experiment subject by existing papers (Zhao et al.

2017; Tian et al. 2020b) and the other two models are included because they are the state-of-the-art in terms of accuracy (till March 2021). All three multi-label classification models are trained on the COCO dataset. Please note that the number of public available multi-label classification models is much smaller than that of the single-label classification models, and we have tried our best efforts to collect these three models.

In the experiment, we found that Keras Application only provided the trained model, but missed the source code to reproduce the results for image classification, especially the code to preprocess the input. To avoid the possible mistakes in reproduction, we leveraged the functionality provided by an open-source toolbox, EvalDNN (Tian et al. 2020a), which has successfully reproduced the reported accuracy for most of the 18 models. The maximum difference between the reported accuracy and the reproduced one is only 0.7%, which demonstrates that we have faithfully deployed the models in our experiments. For the multi-label classification models, we successfully reproduced the results by leveraging the detailed source code provided by the authors.[4] For the threshold in multi-label classification models, we use the value suggested by their documentation, i.e., 0.5 for TResNet-L and TResNet-XL, and 0.7 for ResNet-50. The columns *Reproduced Accuracy* of Table 5 and Table 6 list the accuracy reproduced in this study for single-label classification and multi-label models, respectively. After the deployment, we applied our approach to identify unreliable inferences from all the correct inferences made by these models .

Threats to validity There are two potential threats to validity in this experiment. First, the models used in this experiment may not include all the DNN-based image classification models and our conclusion may have bias. To mitigate the threat, we collected 21 representative and popular models. They covered most of the modern advanced architectures used in image classification. We believe that our conclusions can be generalized. Second, the inference results of these model can be affected due to the mistake in model deployments. To alleviate this threat, we leveraged the existing toolbox (Tian et al. 2020a) and the source code provided by the authors. We ensured that the models deployed in our experiment perform closely to the accuracy reported in their original research publications and documentations.

6.1.3 Results and Discussion

Table 5 and Table 6 show the experimental results of single-label classification models and multi-label classification models, respectively. For each cell, the percentage displayed in the parentheses is the ratio of the number of unreliable inferences found by our approach with respect to the number of the correct inferences. Please note that the column *Inferences Violating MR-1* refers to the number of inferences violating MR-1, regardless of whether MR-2 is violated or not. The column *Inferences Violating MR-2* refers to the number of inferences violating MR-2, regardless of whether MR-1 is violated or not. The last column *Inferences Violating MR-1&2* refers to the number of inferences violating both MR-1 and MR-2.

[4] TResNet-L: `https://github.com/Alibaba-MIIL/ASL`, ResNet-50: `https://github.com/ARiSE-Lab/DeepInspect`

Table 5: Single-Label Image Classification Models, their Accuracy, the Number and Ratio of Unreliable Inferences Violating MRs on the ImageNet dataset.

Model	Reproduced Accuracy	Inferences Violating MR-1	Inferences Violating MR-2	Inferences Violating MR-1&2
Xception	79.0%	374 (0.95%)	4,104 (10.39%)	229 (0.58%)
VGG16	71.3%	259 (0.73%)	5,394 (15.14%)	228 (0.64%)
VGG19	71.3%	252 (0.71%)	5,628 (15.80%)	219 (0.61%)
ResNet50	74.9%	253 (0.68%)	5,248 (14.01%)	197 (0.53%)
ResNet101	76.4%	344 (0.90%)	4,942 (12.93%)	268 (0.70%)
ResNet152	76.6%	334 (0.87%)	4,727 (12.34%)	266 (0.69%)
ResNet50V2	75.3%	247 (0.66%)	5,387 (14.30%)	213 (0.57%)
ResNet101V2	76.9%	271 (0.70%)	4,606 (11.98%)	212 (0.55%)
ResNet152V2	77.7%	319 (0.82%)	4,392 (11.30%)	252 (0.65%)
InceptionV3	77.9%	404 (1.04%)	4,663 (11.98%)	292 (0.75%)
InceptionResNetV2	80.4%	686 (1.71%)	3,998 (9.94%)	388 (0.97%)
MobileNet	70.3%	222 (0.63%)	6,622 (18.83%)	195 (0.55%)
MobileNetV2	71.2%	281 (0.79%)	6,437 (18.08%)	225 (0.63%)
DenseNet121	75.0%	278 (0.74%)	4,349 (11.60%)	219 (0.58%)
DenseNet169	76.2%	340 (0.89%)	4,154 (10.91%)	264 (0.69%)
DenseNet201	77.3%	334 (0.86%)	4,296 (11.11%)	260 (0.67%)
NASNetMobile	73.8%	461 (1.25%)	6,505 (17.64%)	345 (0.94%)
NASNetLarge	82.7%	826 (2.00%)	3,634 (8.79%)	383 (0.93%)

Table 6: Multi-label Image Classification Models, their Accuracy, the Number and Ratio of Unreliable Inferences Violating MRs on the COCO dataset.

Model	Reproduced Accuracy	Inferences Violating MR-1	Inferences Violating MR-2	Inferences Violating MR-1&2
ResNet50	34.5%	657 (4.71%)	4,873 (34.91%)	422 (3.0%)
TResNet-L	45.5%	1,013 (5.49%)	5,028 (27.26%)	442 (2.4%)
TResNet-XL	47.9%	957 (4.93%)	4,732 (24.38%)	362 (1.9%)

The results reveal that each selected single-label and multi-label DNN classification model makes hundreds of unreliable inferences violating MR-1 and thousands of ones violating MR-2. In terms of ratio, for single-label classification, our approach identifies that 0.63%∼2.00% of the correct inferences violate MR-1, and 9.79%∼18.83% of the correct inferences violate MR-2. As for multi-label classification, the ratio is much higher. Specifically, our approach identifies that 4.71%∼5.49% of the correct inferences violate MR-1, and 24.38%∼34.91% of the correct inferences violate MR-2. Furthermore, there are around 2% of the inferences violating both MR-1 and MR-2. The results show that the phenomenon, i.e., model makes inferences based on object-irrelevant features, generally exists across different models.

We further investigated whether different models will make unreliable inferences towards different test inputs. If most of the models make unreliable inferences for the same set of inputs, it is more likely that these inputs are defective. To conduct the investigation, we studied for each input the number of different models whose inference for the input was unreliable. Specifically, the number of different models varies from 1 to N, where N is the total number of models included in our experiments. More specifically, N is 18 for single-label classification on the ImageNet dataset and 3 for multi-label classification on the COCO dataset. We then calculated the ratio of inputs, for which unreliable inferences were made by n models ($n = 1, 2, ..., N$), with respect to the total number of inputs for which unreliable inferences were made by at least one model.

Figure 6a and Figure 6b show the results for single-label classification models on the ImageNet dataset and multi-label classification models on the COCO dataset, respectively. It can be observed that, for single-label classification, 43.7% and 31.8% of the inputs concern unreliable inferences violating MR-1 and MR-2 made by only one model, respectively. More than half of the inputs concern unreliable inferences made by three or fewer models. Only a small portion of inputs (less than 2.7%) concern unreliable inferences made by all 18 models. A similar pattern can also be found for multi-label classification models. 74.3% and 67.4% of the inputs concern unreliable inferences violating MR-1 and MR-2 made by only one model, respectively. Less than 7.8% of the concern unreliable inferences made by all three models.

Such results reveal that different models make unreliable inferences for different sets of inputs, which indicates that such unreliable inferences are more likely to be caused by the models themselves instead of the inputs.

Answer to RQ3 The problem of making unreliable inferences is common to state-of-the-art models. Since these models make unreliable inferences on different input sets, the problem is likely to be caused by models instead of inputs.

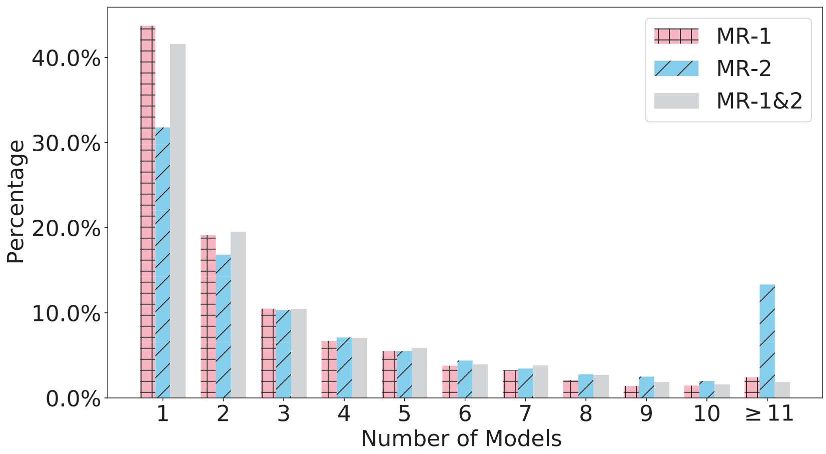
6.2 Characteristic of Unreliable Inferences

RQ4 Is there a correlation between the target object size and the unreliable inferences?

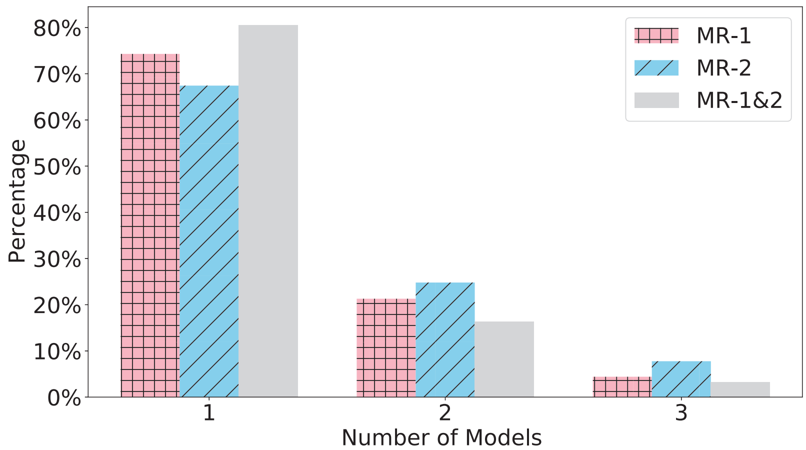
6.2.1 Motivation

As shown in Section 6.1, unreliable inferences are pervasive, and different models make unreliable inferences for different inputs. We are curious about whether common characteristics exhibit among unreliable inferences. If so, we may give some useful suggestions to developers.

We manually investigated those inputs that cause unreliable inferences made by most models. We observed that the sizes of the target objects in these inputs usually occupy a tiny part of the whole image. Figure 7 shows some examples. The objects of these images are different types of balls whose sizes are often small in the images, especially compared to the sports facilities and players. It motivates



(a) Single-Label Classification Models on the ImageNet Dataset



(b) Muti-label Classification Models on the COCO Dataset

Fig. 6: The Percentage of Inputs for which Unreliable Inferences were Made by Different Number of Sinlge-Label and Muti-label Classification Models

us to investigate whether the size of an input's target object is correlated with its probability of being unreliably inferred by DNN models.

6.2.2 Experiment Design

To answer this question, for each unreliable inference, we computed the ratio of the target object's size with respect to the size of the whole input image. Then we divided all inferences into 20 intervals based on their ratios, which are

Fig. 7: The Images that Have Small Objects and Are Unreliably Inferred by DNN Models

$[0.05 * i, 0.05 * (i + 1))$ and i ranges from 0 to 20. For each interval, we computed the ratio of unreliable inferences, with respect to the total number of inferences belonging to this interval. We selected the NASNetLarge and TResNet-XL as experiment subjects since they achieved the highest top-1 accuracy in all models used in our experiment for the ImageNet dataset and the COCO dataset, respectively.

6.2.3 Results and Discussion

Figure 8a and Figure 8b show the results for the model NASNetLarge on the ImageNet dataset and for the model TResNet-XL on the COCO dataset, respectively. Please note that for each interval, we use its middle point as the value in x-axis, except for the last interval we use the point 1.0. We observed that for these inferences whose target objects are smaller (relative to the size of the image), they are more likely to be unreliable. Similar results have been observed among the other models. We suspect that when the model handles an image whose target object is small, it often extracts features from the background region. Eventually, it leverages object-irrelevant features to make decisions.

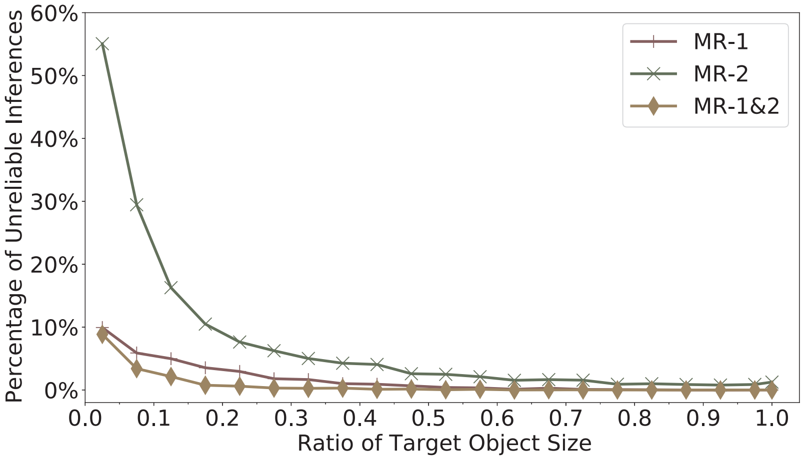
Answer to RQ4 In summary, we found that inputs with small target object sizes are more likely to be unreliably inferred by existing DNN models. We suggest the users of these models to pay more attention when making inferences on these inputs (i.e., the objects' size are less than 30% of the whole image), especially when deploying these models on safety-critical applications.

6.3 Effect of Unreliable Inferences

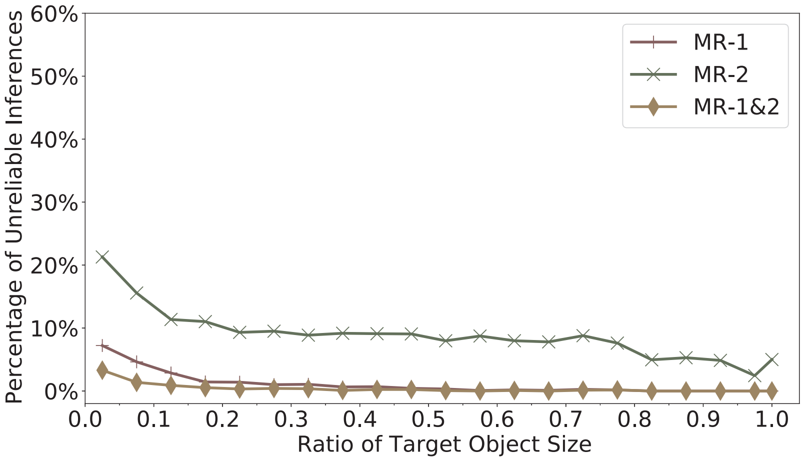
RQ5 To what extent will the unreliable inference affect a model's evaluation?

6.3.1 Motivation

As revealed by previous sections, a significant proportion of the correct inferences made by existing models are unreliable. Such pervasiveness of unreliable inferences might cause bias in understanding and evaluating the performance of different models. Specifically, if there exists a significant amount of unreliable inferences, it could induce non-trivial uncertainties in measuring model accuracy. Therefore,



(a) Single-Label Classification Model NASNetLarge on the ImageNet Dataset



(b) Multi-label Classification Model TResNet-XL on the COCO Dataset

Fig. 8: The Ratio of Unreliable Inferences Made by Single-label and Multi-label Classification Models *w.r.t* the Ratio of Target Object Size

we investigated the effect of unreliable inferences on model accuracy evaluation in this experiment.

6.3.2 Experiment Design

We investigated the effects of unreliable inferences on the measurement of accuracy. Since both the correct and incorrect inferences can be unreliable and both of them are important to model evaluations, in this section, we examined both correct and

Table 7: The Comparison of the Top-1 Accuracy between the Unreliable Inferences and Reliable Inferences for Single-label Image Classification Models

Model	Original	MR-1		MR-2		MR-1&2	
		Unreliable	Reliable	Unreliable	Reliable	Unreliable	Reliable
Xception	79.02%	28.22%	80.42%	45.51%	86.40%	20.00%	80.41%
VGG16	71.27%	21.73%	72.48%	41.10%	82.01%	20.07%	72.46%
VGG19	71.26%	20.62%	72.52%	42.21%	81.82%	18.83%	72.50%
ResNet50	74.93%	21.58%	76.21%	44.98%	84.05%	18.17%	76.19%
ResNet101	76.42%	26.34%	77.76%	45.48%	85.01%	22.48%	77.74%
ResNet152	76.60%	26.13%	77.94%	44.88%	85.07%	22.52%	77.91%
ResNet50V2	75.34%	19.92%	76.78%	46.19%	84.21%	17.82%	76.75%
ResNet101V2	76.89%	21.16%	78.37%	45.05%	85.08%	17.76%	78.34%
ResNet152V2	77.73%	23.19%	79.28%	45.51%	85.44%	19.73%	79.25%
InceptionV3	77.87%	30.01%	79.20%	46.05%	85.95%	24.48%	79.17%
InceptionResNetV2	80.41%	42.52%	81.68%	47.43%	87.10%	30.36%	81.73%
MobileNet	70.34%	21.50%	71.38%	42.84%	82.65%	19.52%	71.37%
MobileNetV2	71.19%	23.44%	72.37%	44.47%	82.08%	20.09%	72.36%
DenseNet121	74.97%	22.32%	76.34%	41.89%	83.64%	18.73%	76.32%
DenseNet169	76.18%	26.19%	77.51%	42.02%	84.59%	22.30%	77.48%
DenseNet201	77.32%	26.05%	78.67%	44.60%	85.13%	22.34%	78.63%
NASNetMobile	73.77%	33.12%	74.94%	46.18%	84.60%	27.38%	74.97%
NASNetLarge	82.68%	46.74%	83.99%	49.44%	88.40%	30.25%	84.04%

incorrect inferences. For the incorrect inferences, it is possible that they have the labels that do not exist in the ground truth and thus the object-relevant features cannot be directly identified. In such cases, we use the union of all the objects in the annotation to approximate the target object and then identify the object-relevant features.

In the investigation, with respect to MR-1, we examined all (both correct and incorrect) inferences and separated them into two sets for each model according to whether they are reliable. One set contains all the inputs whose inferences are identified as unreliable by our approach and another set that contains the remaining test inputs. We denoted the former set as "Unreliable" and denoted the latter one as "Reliable". We also compared the results such obtained with the original accuracy reproduced by our approach, which is denoted as "Original". Similar procedures were applied with respect to MR-2, and the MR-1&2. If the results before and after removing the unreliable inference have a significant difference, it indicates that the unreliable inferences will induce bias for model evaluation. We then re-computed the accuracy based on each set of test inputs and checked if the evaluation results are significantly different by conducting the Wilcoxon signed-rank test (Wilcoxon 1945).

6.3.3 Results and Discussion

Table 7 shows the results aggregated over all the 18 single-label image classification models. In terms of the accuracy evaluated after removing the unreliable inferences with respect to MR-2 (column *MR-2 Reliable*), it is significantly higher than the original accuracy value obtained over all the test inputs (p-value $= 3.81 * e^{-6}$). On average, the model accuracy after removing the unreliable inferences is 8.84%

Table 8: The Comparison of the Top-1 Accuracy between the Unreliable Inferences and Reliable Inferences for Multi-label Image Classification Models

Model	Original	MR-1		MR-2		MR-1&2	
		Unreliable	Reliable	Unreliable	Reliable	Unreliable	Reliable
ResNet50	34.5%	41.66%	34.17%	22.47%	48.29%	33.65%	34.49%
TResNet-L	45.5%	45.98%	45.51%	22.87%	72.45%	28.85%	46.19%
TResNet-XL	47.9%	45.31%	48.06%	23.19%	73.03%	25.84%	48.71%

(5.73%∼12.31%) higher than the original accuracy. For MR-1 and MR-1&2, a certain trend toward significance could also be observed, for which the model accuracy after removing the unreliable inferences is only 1.31% (1.04%∼1.55%) and 1.30% (1.04%∼1.52%) higher than the original accuracy.

Table 8 shows the result of three multi-label classification models. Similar to the previous finding for single-label classification models, after removing the unreliable inferences violating MR-2, the model accuracy is much higher (13.79%∼26.95%) than the original accuracy value obtained over all the test inputs. Please note that the significant test is not applicable since there are only three samples, which is significantly less than 20, the typical minimum number for a significant test.

The above results reveal that the existence of unreliable inferences violating MR-2 causes significant bias for model evaluation, while the effect of unreliable inferences violating MR-1 and MR-1&2 is limited. By excluding those unreliable inferences violating MR-2, the performance of existing models evaluated with respect to accuracy is much higher than that evaluated based on inputs containing unreliable inferences. We suggest developers to remove unreliable inferences for fair model comparisons, especially the inferences violating MR-2.

Besides, in general, as shown in Table 7 and Table 8, the model accuracy on the unreliable inference is significantly lower than the original accuracy of model. However, there are some exceptions. In multi-label image classification (Table 8), the model accuracy on the unreliable inference is higher than (ResNet50 and TResNet-L, MR-1) or close to (ResNet50, MR-1&2 and TResNet-XL, MR-1) the original accuracy of the model. We suggest that the developers should pay more attention to such exceptions: even if the unreliable inferences have a comparable accuracy with the reliable ones, they may raise concerns on model reliability, as we mentioned in Section 1.

Answer to RQ5 The unreliable inferences violating MR-2 can cause significant effects (8.84% for single-label classification and 21.96% for multi-label classification) on the evaluation results, thus inducing bias in model comparisons. On the contrary, the effect of the unreliable inferences violating MR-1 and MR-1&2 is limited.

6.4 Taming Unreliable Inferences

RQ6 Can the unreliable inference be tamed during training?

6.4.1 Motivation

Previous results have shown that unreliable inferences generally exist in widely-used models built with different architectures. Besides, the inputs causing unreliable inferences vary across models. These unreliable inferences can induce significant bias in the evaluation of model performance. In this subsection, we studied whether such unreliable inferences can be tamed. Specifically, our study has two goals.

First, we investigated whether the ratio of unreliable inferences can be reduced during the model training process. Second, we investigated whether there is any correlation between model accuracy and the ratio of unreliable inferences. Understanding their correlation helps formulate a training strategy taming such unreliable inferences. For instance, if the top-1 accuracy is negatively correlated with the ratio of unreliable inference, the ratio of the unreliable inferences is likely to be reduced by enhancing the model accuracy.

6.4.2 Experiment Setup

We conducted two experiments with the aim to achieve the above two goals. First, we trained the VGG16 and Resnet50 models from scratch using the training source code provided by PyTorch official example repository,[5] based on the ImageNet dataset. We selected these two models because they have been popularly adopted by existing studies for testing DNN systems (Pei et al. 2017; Ma et al. 2018a; Tian et al. 2020b; Zhao et al. 2017). The training was based on the default hyper-parameters, and stopped when its accuracy and loss reach saturation. We then measured the ratio of unreliable inferences in all correct inferences for every five epochs during the training process to see if they are reduced. Since the training process of DNN models is stochastic, we repeated the training three times for each model. Please note that the training of these two models is very time-consuming. Although our server has eight 2080Ti GPU cards, it still takes around 80 mins and 30 mins to train one epoch for VGG16 and Resnet50. The total training time spent for this experiment is more than 20 days.

Second, we investigated the correlation between model accuracy and the ratio of unreliable inferences using the pre-trained models in Table 5. Specifically, we used the Pearson Correlation (Benesty et al. 2009) to check whether the ratio of unreliable inferences and the top-1 accuracy are correlated. We also plotted them for visualization.

In this research question, we did not include the multi-label classification due to the following two reasons. First, the source code to train these models is not available. Second, the number of available multi-label classification models is limited and it is not applicable to calculate the Pearson Correlation.

6.4.3 Results and Discussion

On average, our trained VGG16 and Resnet50 models achieve the top-1 accuracy of 72.1% and 76.1%, respectively. Their accuracy is close to the accuracy of the pre-

[5] https://github.com/pytorch/examples

trained models published by Pytorch,[6] which are 71.6% and 76.2%, respectively. Figure 9 shows the top-1 accuracy and the ratio of unreliable inferences during the training stages. Please note the ratios of unreliable inferences violating MR-1&2 are not plotted as they are highly overlapped with the ratios of unreliable inferences violating MR-1.

It can be observed that at the beginning of training, the ratio of unreliable inferences violating MR-2 decreases significantly and the ratio of unreliable inferences violating MR-1 slightly decreases. Later on, both of them become stable with the accuracy becoming saturated. Such results indicate that the current model training methodologies can guide the models to learn object-relevant features to certain extents, as the ratio of unreliable inferences decreases at the first beginning. However, they become less effective with the training epochs increases, as the ratio of unreliable inferences becomes stable after the beginning. In other words, they may not necessarily prevent the model from making unreliable inferences.

We then investigated the correlation between the top-1 accuracy and the ratio of unreliable inferences based on the pre-trained models. The Pearson Correlation coefficients between the ratio of unreliable inferences violating MR-1, MR-2, and MR-1&2 with top-1 accuracy are 0.702, -0.901, and 0.492, respectively. Figure 10 shows the relation of the ratio of unreliable inferences that violate MR and the top-1 accuracy, as well as their linear regression lines. The results indicate a strong negative correlation (-0.901 $<$ -0.9) between the ratio of unreliable inferences violating MR-2 and top-1 accuracy. In other words, higher top-1 accuracy of a model couples with lower ratio of its unreliable inferences violating MR-2. The ratio of unreliable inferences violating MR-1 has a relatively positive correlation with top-1 accuracy. It increases very slightly with the increase in top-1 accuracy. The ratio of unreliable inferences violating MR-1&2 remains about the same. This may be because that the ratio of unreliable inference violating MR-1 and MR-1&2 is relatively small and their changes are not obvious.

Answer to RQ6 The current training methodologies can help the models to reduce the unreliable inference to certain extents, but they become less effective with the training epochs increases and may not necessarily prevent the model from making unreliable inferences.

7 Limitation and Future Work

Our study points out that unreliable inferences commonly exist in the DNN-based image classification models. In this section, we discuss some limitations of our work and the future work. In the future, we will explore the possibility to improve the reliability of inferences made by DNN models and address such unreliable inferences effectively and efficiently.

7.1 Other Possible MRs

We introduced our approach for the MR-1 and MR-2 in Section 4. There are alternative approaches. For example, in the multi-label classification, we consider

[6] https://pytorch.org/docs/stable/torchvision/models.html

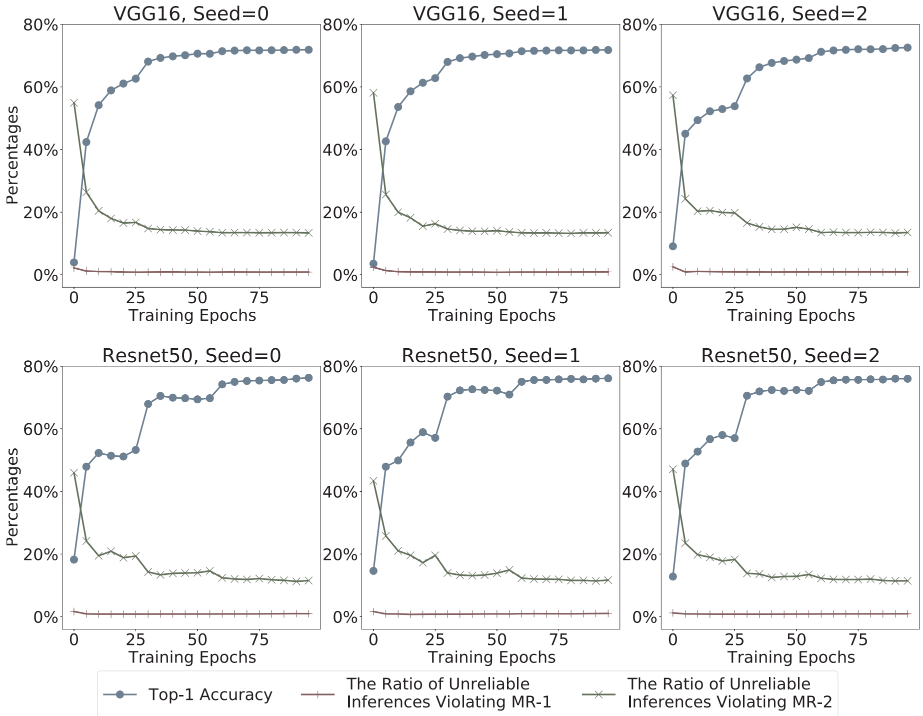


Fig. 9: The Top-1 Accuracy and the Ratio of Unreliable Inferences of VGG16/Resnet50 during Training

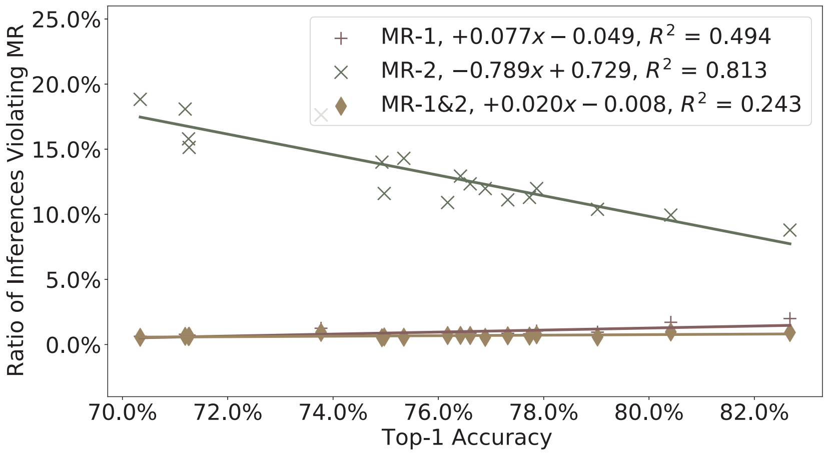


Fig. 10: The Relationship between Top-1 Accuracy and the Ratio of Unreliable Inferences Violating MRs for Single-Label Image Classification Models on the ImageNet Dataset

the union of all the objects holistically and mutate them all together. An alternative way is to consider each label one by one. For example, we only mutate all objects belonging to a specific label at one time and then examine whether this label violates the MR. After examining all labels, one can conclude whether the inference violates the MR. Such an alternative will increase the workload and requires a more sophisticated methodology to judge whether an inference is reliable based on all its labels. We believe there are several potential ways to define such methodology, thus we leave it as future work to conduct an exhausting study.

Further, for multi-label classification, exact match (Wu and Zhu 2020) is used in the comparison of the certainty, i.e., $\mathcal{C}(L_{\mathcal{M}(i)}) > \mathcal{C}(L_{\mathcal{M}(i'_c)}) \iff \mathcal{C}_{l,\mathcal{M}(i)} > \mathcal{C}_{l,\mathcal{M}(i'_c)}, \forall\ l \in\ L_{\mathcal{M}(i)}$. The comparison can use other metrics, such as Hamming Loss and Jaccard Index. In the future, one may investigate the effect of different metrics in the comparison.

7.2 Other Potential Application Scenarios

In our study, we focus on the applications of the DNN on image classification. After proper adaption, our MRs can be applied to other applications used on DNN, such as object detection (Liu et al. 2016; Ren et al. 2017; Redmon et al. 2016) and language processing (Devlin et al. 2019). For example, in object detection, one may examine the object-relevancy for each detected object. The corresponding MRs can be:[7]

MR-3: An image mutated by corrupting only the features of the target object(s) should lead to an inference result with different label(s) and location(s), or an inference result with the same label(s) and location(s) but with less certainty.

MR-4: An image mutated by preserving the features of the target object(s) and corrupting other features should lead to an inference result with the same label(s) and location(s).

As for language processing, the MR could be:

MR-5: A sentence mutated by corrupting only the content words should lead to a different inference result.

MR-6: An sentence mutated by preserving the content words and corrupting other function words should lead to a similar inference result.

Future work can target proposing new MRs for other DNN-based applications and study their effectiveness.

7.3 False Positives and False Negatives

The mutations used in our approach can unnecessarily import/remove extra features and then bring some side effects, such as false positives/negatives. Although we applied three mutation operators and adopted the majority voting to alleviate this threat, it still may happen. In the future work, we will explore different image mutation methods and reduce such possible side effects, including false negatives and false positives.

[7] The MR-3/4/5/6 are just our initial proposals. The detailed definition should be polished and their effectiveness should be thoroughly evaluated.

In our evaluation, we only evaluate the effectiveness of our approach from the perspective of true positives and false positives, but not the false negatives, which are the inferences that are based on the object-irrelevant features but are not detected by our approach. It is challenging to identify the false negatives, since it is hard to know whether the inference is indeed completely based on the object-irrelevant features, which is an outstanding challenge in deep learning (see Section 8.3 and 8.4), and whether the changes of the certainty is caused by the imported/removed features in the mutation. We believe it will be one of the future work directions.

7.4 The Effect of Annotation Formats

Our metamorphic approach leverages the annotation of the object to construct the follow-up inputs. The availability and the quality of the annotation could affect the performance of our approach. This is the major limitation of our study. As shown in the evaluation in Section 5, inappropriate annotations are the major sources of false positives. In the future work, we will explore new methodologies to alleviate this limitation.

In our study, we use bounding boxes for single-label classification and object masks for multi-label classification, depending on their availability in the datasets. We would like to point out that the annotation format could also affect the effectiveness of our approach. For example, if the annotation is in the format of object mask, even after the object corruption in MR-1, the object shape could still be left in the follow-up inputs, which may cause false positives for MR-1 (similar to the incomplete removal of the target object). According to a recent study (Geirhos et al. 2019), the texture of the input image, rather than its shape, has stronger impact in DNN-based image classifications. In other words, "*a cat with an elephant texture is an elephant to CNNs, and still a cat to humans*" (Geirhos et al. 2019). Thus, the influence of the shape information left in the follow-up inputs should be limited. Nevertheless, we would like to point out this possible factor and interested researchers may explore along this direction in the future. A possible countermeasure is to develop a novel mutation methodology such that it will further remove the shape information. For example, we can add random padding to the object boundary, so that the image shape information will be destroyed.

8 Related Work

8.1 Metamorphic Testing in DNN models

Several studies have applied metamorphic testing to validate DNN models (Xie et al. 2011; Ding et al. 2017; Dwarakanath et al. 2018; Zhang et al. 2018; Tian et al. 2018). Dwarakanath et al. (Dwarakanath et al. 2018) leveraged two sets of metamorphic relations to identify faults in machine learning implementations. For example, one metamorphic relation is that the "permutation of input channels (i.e. RGB channels) for the training and test data" would not affect inference results. To validate whether a specific implementation of DNN satisfies this relation, they re-ordered the RGB channel of images in both the training set and test set. They examine the impact on the accuracy or precision of the DNN model after it is trained using the permuted dataset. Their relations treat the pixels in an image as independent units and they do not consider objects and background in the image.

Xie et al. (Xie et al. 2011) performed metamorphic testing on two machine learning algorithms: k-Nearest Neighbors and Naïve Bayes Classifier. Their work targets testing attribute-based machine learning models instead of deep learning systems. Ding et al.(Ding et al. 2017) proposed metamorphic relations for DNN at three different validation levels: system level, data set level and data item level. For example, a metamorphic relation on system level asserts that DNNs should perform better than SVM classifiers for image classification. Their technique requires retraining the systems and is inapplicable to testing pre-trained models.

Other studies (Zhang et al. 2018; Tian et al. 2018; Zhou and Sun 2019) leveraged metamorphic testing to validate autonomous driving systems. DeepTest (Tian et al. 2018) designed a systematic testing approach to detecting the inconsistent behaviors of autonomous driving systems using metamorphic relation. Their relations focus on general image transformation, including scale, shear, rotation and so on. Further, DeepRoad (Zhang et al. 2018) leverages Generative Adversarial Networks to improve the quality of the transformed images. Given an autonomous driving system, DeepRoad mutates the original images to simulate weather conditions such as adding fog to an image. An inconsistency is identified if a DNN model and its mutant make an inconsistent decision on an image (e.g., the difference of the steering degrees exceeds a certain threshold). Differently from the existing study, we design metamorphic relations to assess whether an inference is based on object-relevant features for DNN-based image classification models.

8.2 Testing Deep Learning Systems

Besides metamorphic testing, studies have also been made to adapt other classical testing techniques for DNN models. A recent survey (Zhang et al. 2020) summarizes the latest work in this direction. DeepXplore (Pei et al. 2017) proposed neuron coverage to quantify the adequacy of a testing dataset. DeepGauge (Ma et al. 2018a) proposed a collection of testing criteria. TensorFuzz (Odena et al. 2019), DLFuzz (Guo et al. 2018) and DeepHunter (Xie et al. 2019a) leveraged fuzz testing to facilitate the debugging process in DNN. DeepMutation (Ma et al. 2018b) applied mutation testing to measure the quality of test data in DNN. Our study falls into the research direction of testing DNN systems. One of our major contributions is that we test DNN models from a new perspective, i.e., the object relevancy of inferences.

8.3 Background Dependence of Computer Vision Systems

Some existing work studied the background dependence of computer vision systems, even before the DNN becomes popular (Roobaert et al. 2001; Qin et al. 2010). Qin et al. (Qin et al. 2010) found that removing the background in street scene images can improve the performance of object recognition systems. Rosenfeld et al. (Rosenfeld et al. 2018) demonstrated that after transplanting an object from the training set to the background of another image, the state-of-the-art object detectors could fail to identity the inserted object. Later, Wang and Su (Wang and Su 2020) proposed an automated approach to test the object detectors. Their approach generates test inputs by inserting objects to another image's background. Our study focuses on image classification applications, and we conduct a large-scale empirical study to understand the problem.

8.4 Heatmap-based Testing of DNN Models

Researchers have proposed ideas of generating *HeatMaps* for DNN testing and debugging (Ribeiro et al. 2016; Zhou et al. 2016; Selvaraju et al. 2017; Ma et al. 2018c; Montavon et al. 2019; Fahmy et al. 2020). These HeatMaps essentially capture the *importance* of individual neurons (Ma et al. 2018c) or layers (Montavon et al. 2019; Fahmy et al. 2020) in a given DNN model. Based on different definitions of *importance*, these methods generate different types of HeatMaps. Some of them directly use neuron activation values, gradient values etc. for HeatMap generation (Zhou et al. 2016; Selvaraju et al. 2017). Others perform some extra processing on such raw data, such as calculating the Jacobian matrix or using differential analysis to extract the differences between correctly classified and misclassified samples (Ma et al. 2018c). A common drawback of such methods is that there is no standard definition of neuron/layer importance and it is hard to evaluate whether the generated HeatMaps are correct. As a result, these HeatMaps may or may not accurately reflect neuron/layer importance. Compared to their work, the effectiveness of our approach is properly evaluated.

Moreover, some HeatMap generation techniques require the intermediate information from the models and can only be applied for some specific types of models. For example, CAM (Zhou et al. 2016) and GradCAM (Zhou et al. 2016) requires access to the pooling layer of neural networks, which may not always be available. Different from these methods, our method does not need extra intermediate results from models and thus can be applied to any DNN-based image classification models.

9 Acknowledgment

We want to thank all reviewers for their constructive comments and suggestions for the manuscript. We would also like to thank the editors' coordination. We would like to express our deep gratitude to Miss Yao Feng for her significant contribution to the manual check. Besides, we appreciate the proofreading by our labmates, Mr. Wuqi Zhang, Mr. Meiziniu Li, Mr. Hao Guan and Miss Lei Liu.

This work was supported by the National Key Research and Development Program of China (Grant No. 2019YFE0198100), National Natural Science Foundation of China (Grant No. 61932021, 62002125 and 61802164), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), Hong Kong RGC/RIF (Grant No. R5034-18), Hong Kong ITF (Grant No: MHP/055/19), Hong Kong PhD Fellowship Scheme, MSRA Collaborative Research Grant, Microsoft Cloud Research Software Fellow Award 2019, NSF 1901242, NSF 1910300, and IARPA TrojAI

W911NF19S0012. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

10 Conclusion

In this work, we proposed to leverage metamorphic testing to identify unreliable image classifications made by DNN models based on object-irrelevant features. We proposed two metamorphic relations, from the perspective of object relevancy. We evaluated the effectiveness of our approach and showed that it achieves high precision. We applied our approach to 21 popular pre-trained DNN models with the ImageNet and COCO datasets, and found that the phenomenon of unreliable inferences is pervasive. The pervasiveness caused significant bias in model evaluation. Our experiments revealed that the current model training methodologies can guide the models to learn object-relevant features to certain extent, but may not necessarily prevent the model from making unreliable inferences. Therefore, further research is needed to develop a more effective approach for enhancing a model's object-relevancy property.

References

Aggarwal A, Lohia P, Nagar S, Dey K, Saha D (2019) Black box fairness testing of machine learning models. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, New York, NY, USA, ESEC/FSE 2019, p 625–635, DOI 10.1145/3338906.3338937, URL `https://doi.org/10.1145/3338906.3338937`

Barr ET, Harman M, McMinn P, Shahbaz M, Yoo S (2015) The oracle problem in software testing: A survey. IEEE Transactions on Software Engineering 41(5):507–525

Ben-Baruch E, Ridnik T, Zamir N, Noy A, Friedman I, Protter M, Zelnik-Manor L (2020) Asymmetric loss for multi-label classification. `2009.14119`

Benesty J, Chen J, Huang Y, Cohen I (2009) Pearson correlation coefficient. In: Noise reduction in speech processing, Springer, pp 1–4

Carlini N, Wagner DA (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, IEEE Computer Society, pp 39–57, DOI 10.1109/SP.2017.49, URL `https://doi.org/10.1109/SP.2017.49`

Chen TY, Cheung SC, Yiu SM (1998) Metamorphic testing: a new approach for generating next test cases. Tech. Rep. HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong

Chen TY, Kuo FC, Liu H, Poon PL, Towey D, Tse TH, Zhou ZQ (2018) Metamorphic testing: A review of challenges and opportunities. ACM Comput Surv 51(1):4:1–4:27, DOI 10.1145/3143561, URL `http://doi.acm.org/10.1145/3143561`

Chollet F, et al. (2015a) Keras. `https://keras.io`

Chollet F, et al. (2015b) Keras applications. URL `https://keras.io/api/applications/`

Cochran W (1963) Sampling techniques. 2nd edition. [Wiley Publications in Statistics.], John Wiley & Sons

Cramer H (1946) Mathematical methods of statistics. Princeton University Press, Princeton

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09

Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, pp 4171–4186, DOI 10.18653/v1/n19-1423, URL https://doi.org/10.18653/v1/n19-1423

Ding J, Kang X, Hu X (2017) Validating a deep learning framework by metamorphic testing. In: 2017 IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET), pp 28–34, DOI 10.1109/MET.2017.2

Dwarakanath A, Ahuja M, Sikand S, Rao RM, Bose RPJC, Dubash N, Podder S (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, ACM, New York, NY, USA, ISSTA 2018, pp 118–128, DOI 10.1145/3213846.3213858, URL http://doi.acm.org/10.1145/3213846.3213858

Fahmy H, Pastore F, Bagherzadeh M, Briand L (2020) Supporting dnn safety analysis and retraining through heatmap-based unsupervised learning. 2002.00863

Fellbaum C (2006) Wordnet(s). In: Brown K (ed) Encyclopedia of Language & Linguistics (Second Edition), second edition edn, Elsevier, Oxford, pp 665 – 670, DOI https://doi.org/10.1016/B0-08-044854-2/00946-9, URL http://www.sciencedirect.com/science/article/pii/B0080448542009469

Freund Y, Schapire RE (1995) A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi P (ed) Computational Learning Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 23–37

FRS KP (1900) X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50(302):157–175, DOI 10.1080/14786440009463897, URL https://doi.org/10.1080/14786440009463897

Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W (2019) Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, URL https://openreview.net/forum?id=Bygh9j09KX

Gu T, Liu K, Dolan-Gavitt B, Garg S (2019) Badnets: Evaluating backdooring attacks on deep neural networks. IEEE Access 7:47230–47244, DOI 10.1109/ACCESS.2019.2909068

Guo J, Jiang Y, Zhao Y, Chen Q, Sun J (2018) Dlfuzz: Differential fuzzing testing of deep learning systems. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, New York, NY, USA, ESEC/FSE 2018, p 739–743, DOI 10.1145/3236024.3264835, URL https://doi.org/10.1145/3236024.3264835

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 770–778, DOI 10.1109/CVPR.2016.90

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861, URL http://arxiv.org/abs/1704.04861, 1704.04861

Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, IEEE Computer Society, pp 2261–2269, DOI 10.1109/CVPR.2017.243, URL https://doi.org/10.1109/CVPR.2017.243

Krasin I, Duerig T, Alldrin N, Ferrari V, Abu-El-Haija S, Kuznetsova A, Rom H, Uijlings J, Popov S, Kamali S, Malloci M, Pont-Tuset J, Veit A, Belongie S, Gomes V, Gupta A, Sun C, Chechik G, Cai D, Feng Z, Narayanan D, Murphy K (2017) Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://storagegoogleapiscom/openimages/web/indexhtml

Krizhevsky A, Nair V, Hinton G (2009) The cifar-10 dataset. URL http://www.cs.toronto.edu/~kriz/cifar.html

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ (eds) Advances in Neural Information Processing Systems 25, Curran Associates, Inc., pp 1097–1105, URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33(1):159–174

LeCun Y, Cortes C (2010) MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, URL http://yann.lecun.com/exdb/mnist/

Lin T, Maire M, Belongie SJ, Bourdev LD, Girshick RB, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. CoRR abs/1405.0312, URL http://arxiv.org/abs/1405.0312, 1405.0312

Lin Y, Lv F, Zhu S, Yang M, Cour T, Yu K, Cao L, Huang T (2011) Large-scale image classification: Fast feature extraction and svm training. In: CVPR 2011, pp 1689–1696

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M (eds) Computer Vision – ECCV 2016, Springer International Publishing, Cham, pp 21–37

Ma L, Juefei-Xu F, Zhang F, Sun J, Xue M, Li B, Chen C, Su T, Li L, Liu Y, Zhao J, Wang Y (2018a) Deepgauge: Multi-granularity testing criteria for deep learning systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ACM, New York, NY, USA, ASE 2018, pp 120–131, DOI 10.1145/3238147.3238202, URL http://doi.acm.org/10.1145/3238147.3238202

Ma L, Zhang F, Sun J, Xue M, Li B, Juefei-Xu F, Xie C, Li L, Liu Y, Zhao J, Wang Y (2018b) Deepmutation: Mutation testing of deep learning systems. In: Ghosh S, Natella R, Cukic B, Poston R, Laranjeiro N (eds) 29th IEEE International Symposium on Software Reliability Engineering, ISSRE 2018, Memphis, TN, USA, October 15-18, 2018, IEEE Computer Society, pp 100–111, DOI 10.1109/ISSRE.2018.00021, URL https://doi.org/10.1109/ISSRE.2018.00021

Ma S, Liu Y, Lee WC, Zhang X, Grama A (2018c) Mode: Automated neural network model debugging via state differential analysis and input selection. In: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Association for Computing Machinery, New York, NY, USA, ESEC/FSE 2018, p 175–186, DOI 10.1145/3236024.3236082, URL https://doi.org/10.1145/3236024.3236082

Montavon G, Binder A, Lapuschkin S, Samek W, Müller KR (2019) Layer-wise relevance propagation: an overview. In: Explainable AI: interpreting, explaining and visualizing deep learning, Springer, pp 193–209

Moosavi-Dezfooli S, Fawzi A, Frossard P (2016) Deepfool: A simple and accurate method to fool deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2574–2582, DOI 10.1109/CVPR.2016.282

Nejadgholi M, Yang J (2019) A study of oracle approximations in testing deep learning libraries. In: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp 785–796, DOI 10.1109/ASE.2019.00078

Odena A, Olsson C, Andersen D, Goodfellow IJ (2019) Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, PMLR, Proceedings of Machine Learning Research, vol 97, pp 4901–4911, URL http://proceedings.mlr.press/v97/odena19a.html

Pei K, Cao Y, Yang J, Jana S (2017) Deepxplore: Automated whitebox testing of deep learning systems. In: Proceedings of the 26th Symposium on Operating Systems Principles, ACM, New York, NY, USA, SOSP '17, pp 1–18, DOI 10.1145/3132747.3132785, URL http://doi.acm.org/10.1145/3132747.3132785

Pham HV, Lutellier T, Qi W, Tan L (2019) CRADLE: cross-backend validation to detect and localize bugs in deep learning libraries. In: Proceedings of the 41st International Conference on Software Engineering, IEEE Press, ICSE '19, p 1027–1038, DOI 10.1109/ICSE.2019.00107, URL https://doi.org/10.1109/ICSE.2019.00107

Qin G, Vrusias B, Gillam L (2010) Background filtering for improving of object detection in images. In: 2010 20th International Conference on Pattern Recognition, pp 922–925, DOI 10.1109/ICPR.2010.231

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788, DOI 10.1109/CVPR.2016.91

Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6):1137–1149, DOI 10.1109/TPAMI.2016.2577031

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, San Francisco, CA, USA, August 13-17, 2016, pp 1135–1144

Roobaert D, Zillich M, Eklundh J (2001) A pure learning approach to background-invariant object recognition using pedagogical support vector learning. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol 2, pp II–II, DOI 10.1109/CVPR.2001.990982

Rosenfeld A, Zemel RS, Tsotsos JK (2018) The elephant in the room. CoRR abs/1808.03305, URL `http://arxiv.org/abs/1808.03305`, `1808.03305`

Sanchez J, Perronnin F (2011) High-dimensional signature compression for large-scale image classification. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, USA, CVPR '11, p 1665–1672, DOI 10.1109/CVPR.2011.5995504, URL `https://doi.org/10.1109/CVPR.2011.5995504`

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: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, IEEE Computer Society, pp 618–626, DOI 10.1109/ICCV.2017.74, URL `https://doi.org/10.1109/ICCV.2017.74`

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Bengio Y, LeCun Y (eds) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings

Stock P, Cissé M (2018) Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In: Ferrari V, Hebert M, Sminchisescu C, Weiss Y (eds) Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI, Springer, Lecture Notes in Computer Science, vol 11210, pp 504–519, DOI 10.1007/978-3-030-01231-1_31, URL `https://doi.org/10.1007/978-3-030-01231-1_31`

Tian Y, Pei K, Jana S, Ray B (2018) Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In: Proceedings of the 40th International Conference on Software Engineering, ACM, New York, NY, USA, ICSE '18, pp 303–314, DOI 10.1145/3180155.3180220, URL `http://doi.acm.org/10.1145/3180155.3180220`

Tian Y, Zeng Z, Wen M, Liu Y, Kuo Ty, Cheung SC (2020a) Evaldnn: A toolbox for evaluating deep neural network models. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings, Association for Computing Machinery, New York, NY, USA, ICSE '20, p 45–48, DOI 10.1145/3377812.3382133, URL `https://doi.org/10.1145/3377812.3382133`

Tian Y, Zhong Z, Ordonez V, Kaiser G, Ray B (2020b) Testing dnn image classifiers for confusion & bias errors. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, Association for Computing Machinery, New York, NY, USA, ICSE '20, p 1122–1134, DOI 10.1145/3377811.3380400, URL `https://doi.org/10.1145/3377811.3380400`

Tramèr F, Atlidakis V, Geambasu R, Hsu D, Hubaux J, Humbert M, Juels A, Lin H (2017) Fairtest: Discovering unwarranted associations in data-driven applications. In: 2017 IEEE European Symposium on Security and Privacy (EuroS P), pp 401–416, DOI 10.1109/EuroSP.2017.29

Wang S, Su Z (2020) Metamorphic object insertion for testing object detection systems. In: Proceedings of the 35th ACM/IEEE International Conference on Automated Software Engineering, ACM, New York, NY, USA, ASE 2020, pp 1053–1065, DOI 10.1145/3324884.3416584, URL `http://doi.acm.org/10.1145/3324884.3416584`

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics Bulletin 1(6):80–83

Wu G, Zhu J (2020) Multi-label classification: do hamming loss and subset accuracy really conflict with each other? In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, URL `https://proceedings.neurips.cc/paper/2020/hash/20479c788fb27378c2c99eadcf207e7f-Abstract.html`

Xie X, Ho JW, Murphy C, Kaiser G, Xu B, Chen TY (2011) Testing and validating machine learning classifiers by metamorphic testing. Journal of Systems and Software 84(4):544 – 558, DOI https://doi.org/10.1016/j.jss.2010.11.920, URL `http://www.sciencedirect.`

com/science/article/pii/S0164121210003213, the Ninth International Conference on Quality Software

Xie X, Ma L, Juefei-Xu F, Xue M, Chen H, Liu Y, Zhao J, Li B, Yin J, See S (2019a) Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In: Zhang D, Møller A (eds) Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2019, Beijing, China, July 15-19, 2019, ACM, pp 146–157, DOI 10.1145/3293882.3330579, URL https://doi.org/10.1145/3293882.3330579

Xie X, Ma L, Wang H, Li Y, Liu Y, Li X (2019b) Diffchaser: Detecting disagreements for deep neural networks. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, pp 5772–5778, DOI 10.24963/ijcai.2019/800, URL https://doi.org/10.24963/ijcai.2019/800

Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Generative image inpainting with contextual attention. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, pp 5505–5514, DOI 10.1109/CVPR.2018.00577

Zhang JM, Harman M, Ma L, Liu Y (2020) Machine learning testing: Survey, landscapes and horizons. IEEE Transactions on Software Engineering pp 1–1, DOI 10.1109/TSE.2019.2962027

Zhang M, Zhang Y, Zhang L, Liu C, Khurshid S (2018) Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ACM, New York, NY, USA, ASE 2018, pp 132–142, DOI 10.1145/3238147.3238187, URL http://doi.acm.org/10.1145/3238147.3238187

Zhang P, Wang J, Sun J, Dong G, Wang X, Wang X, Dong JS, Ting D (2020a) White-box fairness testing through adversarial sampling. In: Proceedings of the 42nd International Conference on Software Engineering, Association for Computing Machinery, New York, NY, USA, ICSE '20

Zhang X, Xie X, Ma L, Du X, Hu Q, Liu Y, Zhao J, Sun M (2020b) Towards characterizing adversarial defects of deep learning software from the lens of uncertainty. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, Association for Computing Machinery, New York, NY, USA, ICSE '20, p 739–751, DOI 10.1145/3377811.3380368, URL https://doi.org/10.1145/3377811.3380368

Zhao J, Wang T, Yatskar M, Ordonez V, Chang KW (2017) Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp 2941–2951, URL https://www.aclweb.org/anthology/D17-1319

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2921–2929, DOI 10.1109/CVPR.2016.319

Zhou ZQ, Sun L (2019) Metamorphic testing of driverless cars. Commun ACM 62(3):61–67, DOI 10.1145/3241979, URL http://doi.acm.org/10.1145/3241979

Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8697–8710, DOI 10.1109/CVPR.2018.00907