# Background Invariance Testing According to Semantic Proximity

Zukang Liao
University of Oxford
zukang.liao@eng.ox.ac.uk

Min Chen
University of Oxford
min.chen@oerc.ox.ac.uk

## Abstract

*In many applications, machine-learned (ML) models are required to hold some invariance qualities, such as rotation, size, and intensity invariance. Among these, testing for background invariance presents a significant challenge due to the vast and complex data space it encompasses. To evaluate invariance qualities, we first use a visualization-based testing framework which allows human analysts to assess and make informed decisions about the invariance properties of ML models. We show that such informative testing framework is preferred as ML models with the same global statistics (e.g., accuracy scores) can behave differently and have different visualized testing patterns. However, such human analysts might not lead to consistent decisions without a systematic sampling approach to select representative testing suites. In this work, we present a technical solution for selecting background scenes according to their semantic proximity to a target image that contains a foreground object being tested. We construct an ontology for storing knowledge about relationships among different objects using association analysis. This ontology enables an efficient and meaningful search for background scenes of different semantic distances to a target image, enabling the selection of a test suite that is both diverse and reasonable. Compared with other testing techniques, e.g., random sampling, nearest neighbors, or other sampled test suites by visual-language models (VLMs), our method achieved a superior balance between diversity and consistency of human annotations, thereby enhancing the reliability and comprehensiveness of background invariance testing.*

## 1. Introduction

There are a variety of invariance qualities associated with machine-learned models. Testing these invariance qualities enables us to evaluate the robustness of a model in its real-world application, where the model may encounter variations that do not feature sufficiently in the training and testing data. Testing also allows us to observe possible biases or spurious correlations that may have been learned by a model

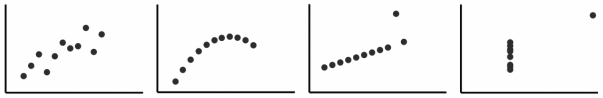

Figure 1. The four sets of points, i.e., the Anscombe's quartet [36], have exactly the same statistical measures, e.g., mean, standard deviation, correlation, etc. However, they are differently distributed. Visualization-based approaches are often more informative than statistical scores for illustrating data distributions.

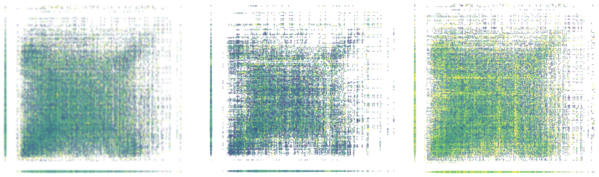


Figure 2. Random sampling leads to inconsistent visual representations across different testing runs, making visualization-based testing frameworks or human-centric testing methods inconsistent. We show the proposed testing framework leads to more consistent testing patterns in Appendix B4.

[39] and to anticipate whether the model can be deployed in other application domains [38]. This work is concerned with background invariance testing – a relatively challenging type of testing.

When considering invariance qualities of an ML model, most existing works reported a single averaged worst-case accuracy, e.g., [15, 40]. However, as shown in Figure 1, typical statistical measures, e.g., mean, cannot informatively characterize how the data are distributed. In machine learning (ML), while averaged accuracy scores can provide an overall statistical indication of the invariance quality, they do not support more detailed analysis such as whether the level of robustness or biases is acceptable in an application by taking into account the probabilities of the variants, for example different background scenes. To transform the problem of invariance testing from simply reporting an accuracy score into a more informative multi-factor decision-making process, recent work [12, 25] conducted testing for

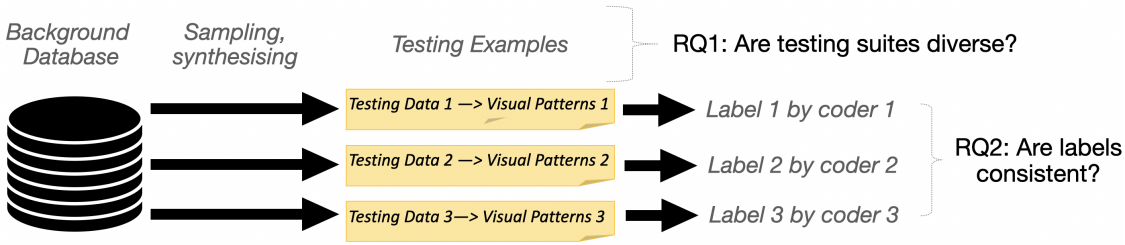


Figure 3. Research questions: to utilize informative visualization-based techniques for background invariance testing, this paper focused on: 1) are the selected testing examples diverse, and 2) are the resultant human decisions (based on the visual patterns) consistent.

basic invariance qualities, e.g., rotation etc, based on visual matrices (formed using all testing results). In this work, we show that ML models with the same accuracy and worst-case accuracy score can exhibit different visualized testing patterns. Therefore, we believe a visualization-based testing framework is preferred to conduct more informative tests.

However, visualization-based testing frameworks often rely on human judgment of visualized patterns. When working with a vast data space, selecting only a small subset for evaluation without a systematic sampling strategy can result in inconsistent test-example selections across different runs. This inconsistency leads to shifting visual patterns and unstable human judgments, as illustrated in Figure 2. Conversely, a sampling strategy that favors a specific type of test case may yield more consistent human judgments but at the cost of reduced diversity in the test suite. To effectively leverage visualization-based testing (more informative) for background invariance, this paper aims to balance the trade-off between test data diversity and judgment consistency (as shown in 3). Our contributions are:

a. We qualitatively confirm that the visualization-based testing method is more informative by showing that ML models with the same averaged worst-case accuracy score can behave differently (i.e., different visual patterns).
b. We qualitatively confirm that the visualization-based testing method suffers from trade-offs between diversity of testing suites and consistency between human decisions.
c. To overcome the trade-off, we introduce an algorithm to search for $n$ desired background scenes based on the semantics encoded in each original image using association ontology.
d. We quantitatively prove that our testing approach is the most balanced between diversity (*recall*) and consistency (*precision*) with the highest f1 score, compared with other testing methods.
e. We show that the proposed background invariance testing based on visual representations can be fully automated.

## 2. Related Works

Invariance qualities of ML models have been studied for a few decades. In recent years, invariance testing has become a common procedure in invariant learning [2, 6, 34]. Among different invariance qualities, background invariance is attracting more attention. In the literature, several types of variations were introduced in background invariance testing, e.g., by replacing the original background with random noise, color patterns, and randomly selected background images. In this section, firstly, we introduce existing attempts, followed by an introduction to other techniques that are used in this work to help conduct background invariance testing.

Rosenfeld, et al. [33] tested object detection models by transforming the original background to random noise or black pixels. They reported that all tested models failed to perform correctly at least in one of their testing cases. Similarly, others, e.g., [4, 5, 42], replaced parts of the images with black or gray pixels for foreground invariance testing. [8] noticed that the association between a foreground object and its background scene affected object recognition and described such association as "consistency". Lauer and Cornelissen [20] tested different models with consistent and inconsistent backgrounds, while using the term "semantically-related" to describe consistent association. In particular, they used color texture to replace the original background of the target image and controlled the inconsistency using a parameterized texture model [32]. Several researchers experimented with swapping background scenes in studying background invariance, e.g., [8]. Xiao, et al. [40] provided the Background Challenge database by overlaying a foreground object to all extracted backgrounds from other images. To prepare models (to be tested), they also provided a smaller version of ImageNet with nine classes (IN9). In this work, we train a small repository of models on IN9, i.e., the models being tested in this work were trained for image classification.

To test deep models, an ideal testing suite should trigger as many neurons as possible. Pei et al. [31] defined the percentage of neurons triggered by a testing suite as the *neuron coverage rate* of the testing suite. Recent studies utilize coverage-based fuzzing techniques to find a testing suite that triggers more potential "bugs" for an ML model, e.g., TensorFuzz [28] and DeepHunter [41]. In this work, we

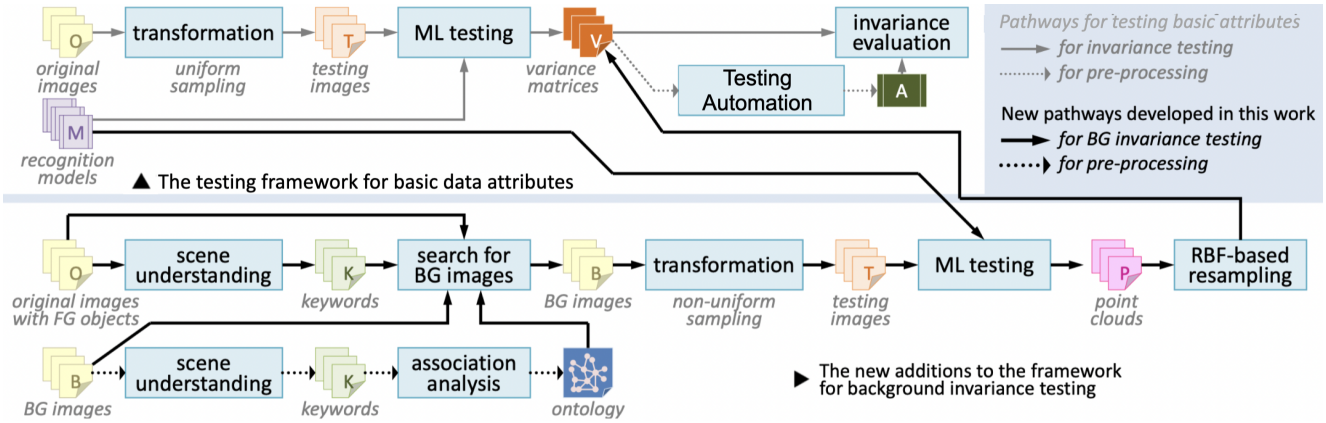


Figure 4. The upper part of the figure shows the invariance testing framework for simple data attributes, e.g., rotation, where the transformations for invariance testing are uniformly sampled. As the transformations for background invariance testing cannot be easily sampled in a consistent way, we introduce a new sub-workflow (lower part) with an additional set of technical components to enable non-uniform sampling of such transformations. This sub-workflow is detailed in Section 4 Methodology. All the trained models and datasets are available at https://github.com/Zukang-Liao/background_invariance_testing.

adopt the most commonly used neuron coverage rate [31] to evaluate the synthesized testing images; Higher neuron coverage rates indicate more diverse testing images [19]. For these reasons, in this work, we use neuron coverage rate to indicate how diverse and comprehensive a testing suite is.

More recent works on invariance testing found that averaged accuracy scores are not informative enough to judge the performance of ML models [7, 9]. Instead, visualized representations are becoming more popular for conducting more informative tests and analyses. For example, Engstrom et al. [12] used 3D heatmaps to analyze the translation/rotation invariance qualities. Liao et al. [25] used visualization to depict different model behaviors for which statistics cannot. Liao and Cheung [23] showed that analyzing the invariance qualities based on visualized patterns can achieve a higher inter-rater reliability (IRR) score than many NLP tasks, confirming the plausibility of conducting invariance testing based on visual patterns. In this work, we show that our visualization-based background invariance testing framework can lead to a satisfactory IRR score with the assistance of our novel technical components, e.g., ontology built on association analysis.

## 3. Definition, Overview, and Motivation

Let $\mathbf{x}_i$ be the $i^{th}$ image in a dataset $D$, $o_i$ be the foreground object in $\mathbf{x}_i$, and $\mathbf{M}$ be an ML model trained to recognize or classify $o_i$ from $\mathbf{x}_i$. In general, the invariance quality of $\mathbf{M}$ characterizes the ability of $\mathbf{M}$ to perform consistently when a type of transformation is applied to $\mathbf{x}_i$. For example, one may apply a sequence of rotation transformations $\mathbf{y}_{i,j} = R(\mathbf{x}, j^\circ),\ j = 0, 1, \ldots$, and test $\mathbf{M}$ with the newly transformed/generated testing images of $\mathbf{y}_{i,j}$. When the testing results can easily be sampled and organized, the visual patterns can facilitate detailed human analysis, e.g., whether the level of robustness or biases is acceptable when taking into account the probabilities of the variants.

The background invariance quality characterizes the ability of $\mathbf{M}$ in recognizing $o_i$ when it is with different backgrounds. Hence the transformations of $\mathbf{x}_i$ involve the replacement of the original background in $\mathbf{x}_i$ with different background scenes $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$. The transformations:

$$\mathbf{y}_{i,j} = Mask_i \circledast x_i + (1 - Mask_i) \circledast \mathbf{b}_j,\ j = 1, 2, \ldots, n \quad (1)$$

where $Mask_i \circledast x_i = o_i$. However, selecting a meaningful and suitable testing suite from an enormous data space is difficult. Therefore, background invariance testing is often carried out based on random selection, resulting in meaningless visual representations and unreliable human judgments (Figure 2). If we can find a way to consistently produce visual representations for the testing results from background transformations, we should be able to conduct visual analysis and the judgment on the background invariance qualities should be consistent and reliable. This motivates us to address the following challenges:

1. to have an effective way to search for background scenes that will be distributed appropriately to form meaningful visualized testing results.
2. to show that the selected background scenes are diverse, broad, and representative.
3. to show that the visual representations are meaningful and the judgments based on them are reliable.
4. to show that such testing procedure can also be automated, therefore it becomes less labor-intensive and more time-efficient.

In this work, we introduce a number of technical components (lower part of Figure 4) to address the aforemen-

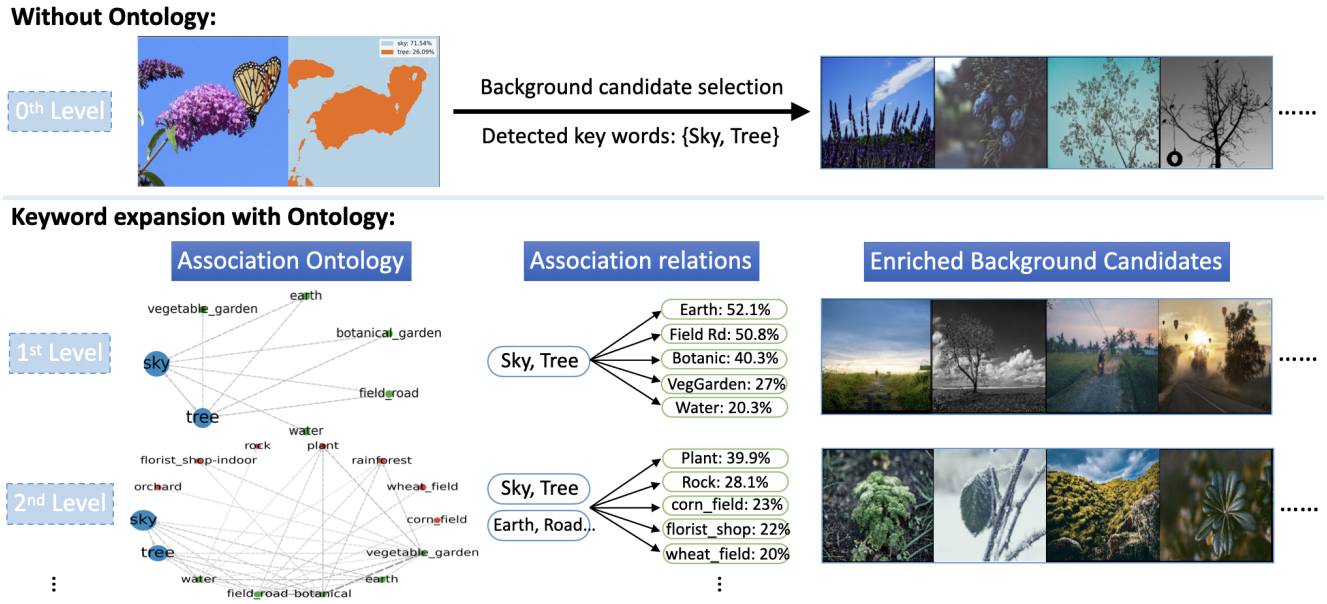


Figure 5. An example image has only two keywords detected by a pre-trained scene understanding model, namely $\{sky, tree\}$. Using an ontology, more keywords can be discovered iteratively, increasing the number and diversity of background scenes.

tioned challenges. Once these challenges are addressed, background invariance testing can be improved from reporting a single average accuracy score to a consistent and reliable judgment based on tailored visual representations.

## 4. Methodology

In this section, we follow the pathways in the lower part of Figure 4 to describe a series of technical solutions for enabling background invariance testing with non-uniform sampling of the transformations of the original images.

### 4.1. To Obtain Detected and Expanded Keywords

In prior works, background candidates were sampled randomly, leading to inconsistent visual representations. In this work, we introduce a systematic sampling approach to obtain background testing candidates. Firstly, we use a scene understanding model and association analysis to build an ontology. We then use the ontology to retrieve indirectly relevant keywords to the original images. Finally, with a set of detected and expanded keywords, keyword-based sampling is used to obtain background testing candidates.

#### 4.1.1. Detected Keywords (Scene Understanding)

From each testing image, a scene understanding model identifies a set of objects that are recorded as a set $K_a$ of keywords. Using the keywords extracted from each image, multiple keywords can be identified for most images, but in many cases, fewer than 3 keywords were detected. We list the statistics in Figure 9 and show some examples in Figure 18 in the Appendix A. To select a suitable (non-random) testing suite that contains rich semantics, it is desirable to consider not only the original keywords, but also other keywords that are related to the detected keywords.

#### 4.1.2. Expanded Keywords: Association Ontology

To consider indirectly relevant keywords to the original testing image, we built an ontology using association analysis, i.e., Apriori algorithm [1] and Frequent Pattern Growth algorithm [16]. Given a set of all possible keywords $K_{\text{all}}$ that can be extracted from all images in a dataset $\mathbb{B}$, the level of association between two keywords $k_a$ and $k_b$ can be described by *support* and *confidence*. For three itemsets: $s_a = \{k_a\}$, $s_b = \{k_b\}$, and $s_{ab} = \{k_a, k_b\}$, the *support* for the itemset $s_{ab}$ is defined as:

$$support(s_{ab}) = \frac{\text{number of images where } s_{ab} \text{ is present}}{\text{total number of images}}$$

which indicates the co-occurrence rate of $k_a$ and $k_b$.

An association rule from one itemset to another, denoted as $\exists s_a \rightarrow \exists s_b$, is defined as *confidence*:

$$confidence(\exists s_a \rightarrow \exists s_b) = \frac{support(s_a \bigcup s_b)}{support(s_a)} \quad (2)$$

which indicates the confidence level about the inference that if the object of keyword $k_a$ appears in a scene, the object of keyword $k_b$ could also appear in such a scene. Similarly, we can compute $confidence(\exists s_b \rightarrow \exists s_a)$. For some non-hierarchical specific keyword, the value of $support(s_{1,2})$ is usually tiny, and is more easily changed by the increase of images in the repository, the introduction of more keywords,

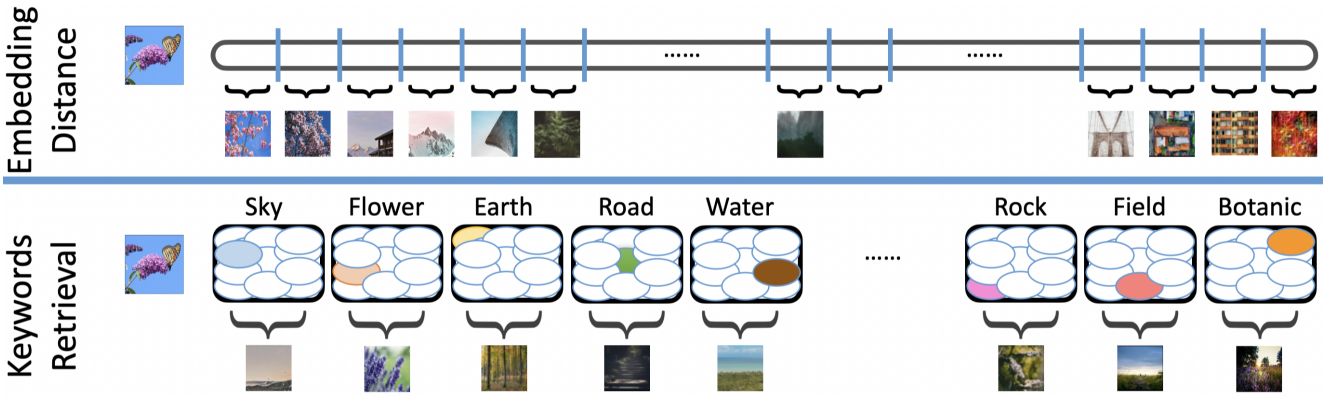


Figure 6. We can sample background scenes based on: 1) similarity/distance metrics, or 2) keywords. When sampling one instance from each subspace, we choose the first one leads to a higher realness score defined by [14] than a threshold.

and the improvement of scene understanding techniques. We therefore use *confidence* as the weights (directed edges) in our ontology.

In the ontology, the shortest path between two keywords indicates the level of association between them, typically facilitating two measures, i) the number of edges along the path (i.e., hops) and ii) an aggregated weight, e.g., $\prod_i^h w_{i=1}$ or $\min(0, w_1 - \sum_{i=2}^{h}(1-w_i)^{\alpha_i}(\alpha_i \geq 1))$. As illustrated in Figure 5, nodes represent keywords, and an edge between two nodes indicates that two keywords have been detected from the same image at least once. The weight on the edge indicates how strong is the association between the two keywords. The ontology is typically constructed in a pre-processing step by training association rules using the extracted keywords for all images in a dataset.

A set of keywords $K_x$ extracted by a scene understanding model can be used to search for background scenes with at least one of the matching keywords $k \in K_x$. When there are many keywords in $K_x$, search based on original keywords can work very well. However, as exemplified in Figure 5(top), when an image has only two keywords, the search will likely yield a small number of background scenes, undermining the statistical significance of the test.

To address this issue, we expand the keyword set $K_x$ by using the ontology that has acquired knowledge about keyword relationships in the preprocessing stage. As illustrated in Figure 5, the initial set $K_x$ has keywords [sky, tree]. The ontology shows that {Sky, Tree} are connected to {Earth, Field Road, Botanic Garden, Vegetable Garden, Water}, which form the level 1 expansion set $E_{1,x}$. Similarly, from $E_1$, the ontology helps us to find the level 2 expansion set $E_{2,x}$, and so on. The set of all keywords after $i$-th expansion is:

$$\text{OL}_x[i] = K_x \cup \left( \bigcup_{j=1}^{i} E_{i,x} \right) \tag{3}$$

## 4.2. To Synthesize/Generate Testing Images

For an original image, with a set of keywords (detected and expanded), one can synthesize testing images by i) generative blending, or ii) simple background replacement Eq. 1. In this work, we use the latter to avoid unwanted foreground objects. We guarantee that no foreground objects will appear in any background scene by carefully selecting the dataset of the background candidates.

### 4.2.1. Background Scenes Sampling

Given a target image $\mathbf{x}$, to test if an ML model is background-invariant, we need to sample a set of background scenes that can be used to replace the original background in $\mathbf{x}$ while maintaining the foreground object $o$. As shown in Figure 6, to replace random sampling, we can sample background scenes based on: $\text{a}_1$) cosine/l2 distance between embeddings of the original image and testing images, or $\text{a}_2$) keywords. When (randomly) sampling one instance from a subspace defined by $\text{b}_1$) a distance interval (bin), or $\text{b}_2$) background scenes containing a certain keyword, we run the Dreamsim [14] model to select the first background scene that leads to a testing image with a higher realness score than a threshold. In Section 5, we quantitatively show that keyword-based sampling is the most balanced between diversity and reliability.

### 4.2.2. Simple Background Replacement

To test ML models that were trained to recognise or classify foreground objects, any background scenes containing any of the foreground objects should not be used. For this reason, we did not use generative algorithms to do the background replacement. We show some generated testing images using the latest generative model (blended latent diffusion [3]) in Figure 7. Some generated images include foreground objects due to unwanted biases, e.g., fish in the sea. This will affect the models' behaviors in an uncontrolled way. Furthermore, running large generative models

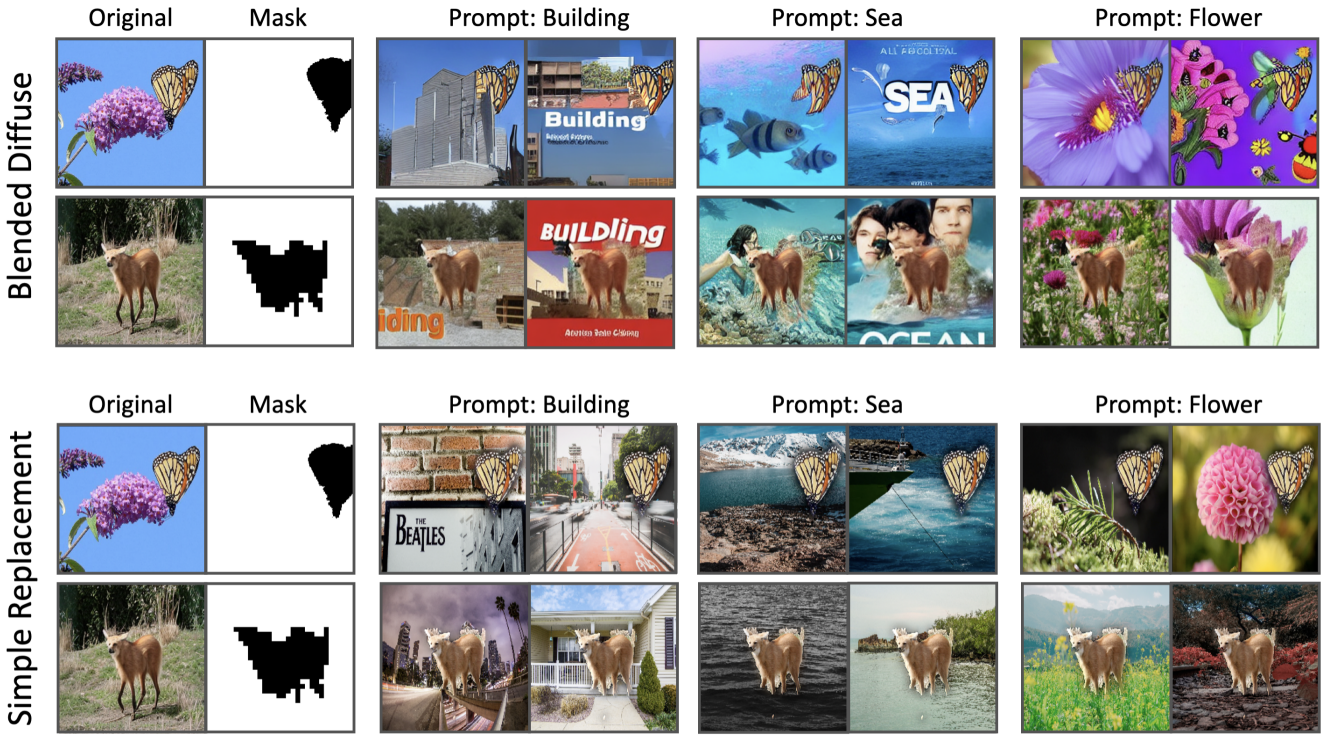


Figure 7. Background invariance testing images generated by the blended diffuse model [3] (first row), and synthesized by simple background replacement (a segmentation model [44], second row). The generated images can be more realistic, i.e., smooth blending. However, generative models can include unwanted biases, e.g., fish in the sea and insect on a flower, both of which are one of the nine foreground objects that the original models (to be tested) were trained to classify.

can be computationally expensive, taking hours to generate one testing image. For these two reasons, we use simple background replacement Eq. 1 together with image blending using Laplacian pyramids [30] to remove some artifacts.

### 4.3. To Analyse Testing Results

After we have synthesized testing images, we can 1) evaluate the diversity of the testing suite, 2) visualize the testing results to analyse the target ML model, and 3) examine if human judgements and decisions based on the visualized testing results are consistent and reliable.

#### 4.3.1. Diversity of Selected Testing Images

We evaluate the diversity and comprehensiveness using the neuron coverage rate [31] (percentage of triggered neurons) which has been commonly used to evaluate the extensiveness of a testing suite for ML models [28, 41]. A lower neuron coverage rate often indicates monotonicity, whereas a higher coverage rate often indicates that the testing suite is sufficiently diverse, engaging a broad spectrum of the targeted ML model's internal structures [18, 19].

#### 4.3.2. Visualization of Testing Results

When we test an ML model $\mathbf{M}$ (trained for image classification) against the testing images, we can measure the results and intermediate results of $\mathbf{M}$ in many different locations. To reduce the number of locations (neurons) to be tested, we select the final predictions (confidence scores), and the embeddings after the final pooling layer. For background invariance testing, we can further reduce the number of neurons to be tested by utilizing the mask of the foreground objects. We feed the foreground-only image $o_i$ into $\mathbf{M}$, and obtain the top k neurons in the embedding layer. For these top $k$ neurons, we can investigate each of them or their statistics, e.g., mean or max. After we have decided the testing positions $ps$, and we can collect the response signal from $\mathbf{M}$ given a testing image $\mathbf{y}_{i,j}$, we denote the response signal as $S(\mathbf{M}, ps, \mathbf{y}_{i,j})$.

Meanwhile, we measure the *semantic distance* between each testing image $\mathbf{y}_{i,j}$ and the original image $\mathbf{x}_i$ using an ensemble of ViT, CLIP, ResNet, and VGG models, which was designed for semantic image similarity [22]. Consider two different testing images $\mathbf{y}_{i,j}$ and $\mathbf{y}_{i,k}$ and their corresponding semantic distances to $\mathbf{x}_i$ as $d_{i,j}$ and $d_{i,k}$. The difference between their numerical measures

$$v_{j,k} = \text{dif}(S(\mathbf{M}, ps, \mathbf{y}_{i,j}), S(\mathbf{M}, ps, \mathbf{y}_{i,k})) \quad (4)$$

indicates the variation between the two testing results. As the variation corresponds to positions $d_{i,j}$ and $d_{i,k}$, this

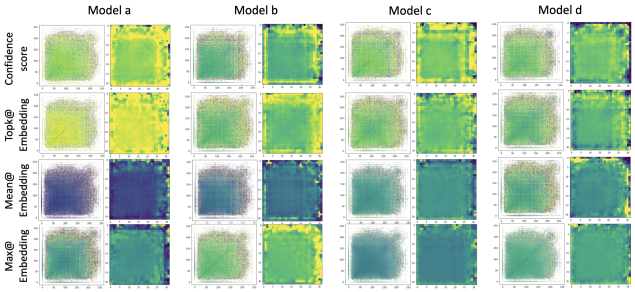


Figure 8. We show four trained ML models on IN9 dataset have different visualized testing results (of the same test run) even though they have the same accuracy score and worst-case accuracy score (averaged). We analyse these patterns in Section 5.1.

gives us a 2D data point at coordinates $p_{j,k} = (d_{i,j}, d_{i,k})$ with data value $v_{j,k}$. When we consider all the testing results for all $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,n}$ as well as $\mathbf{x}_i$, there is point cloud with $n(n+1)$ data points in the context of $\mathbf{x}_i$.

When we combine the testing results for all $l$ targeting images, we have a point cloud with $ln(n+1)$ data points, which can be visualized as scatter plots. To overcome the visualization problem of overlapping glyphs for dense areas, we adopted a common approach of radial basis functions (RBF) to transform point clouds (left column in Figure 8) into a variance matrix (right column in Figure 8). We detailed the adjusted RBF algorithm in Appendix D.

#### 4.3.3. Human Decisions and Annotations

For one targeted ML model, based on the visualized testing results, we ask three ML practitioners to annotate if the model is: a) background invariant, b) borderline, or c) not invariant. For a model repository of 250 ML models, we evaluate the reliability and consistency of the human annotations by the inter-rater reliability [11] (IRR) between the three annotators.

#### 4.3.4. Comparing Different Testing Methods

Although retrieving all similar background candidates to the original image $\mathbf{x}_i$ could lead to a high consistency (IRR) of human annotations, it might not always be desired because the selected testing suite will only include similar scenarios. The ideal selected testing method should include:

- diverse and comprehensive testing suites (*recall*).
- consistent and reliable human judgements (*precision*).

In this work, we compare different testing methods, including 1) different background candidate sampling approaches (random, interval, or keyword-based), and 2) different keyword selection approaches (association ontology or vision language models, e.g., CLIP), by reporting f1 scores. We show that our method (keyword-based sampling using ontology) is the most balanced between diverse testing suites and reliable human judgements, leading to the best f1 score.

## 5. Experiments

To prepare ML models to be tested, we train a small repository of 250 models on the IN9 database (a smaller ImageNet with only nine classes) [40]. To guarantee that no foreground objects would appear in our testing images, we use the BG-20k [21] database with 20,000 background scenes as all the background candidates. We select $N = 32$ background scenes for each target image, and retrieve background scenes using the keyword-based sampling or distance-based sampling methods described in Section 4.2.1. We show that similar results can be found when $N = 50$ or $N = 100$ in Appendix B.

For each model, we measure the signals at two positions, including the final predictions (confidence score) and the embedding layer (after the final pooling layer, or the last layer before the final MLP for Vision Transformers), as these two positions are considered the most important and interesting by our annotators. For the embedding layer, we choose a) the average value of the top 3 neurons triggered by foreground-only images, b) the maximum value, and c) the average value. This leads to one scatter plot for the confidence score and three scatter plots from the embedding layer. For each model, we provide professional annotations based on the visualized testing results with three quality levels, namely, 1) not invariant, 2) borderline, and 3) invariant. In Appendix C, we show the statistics of the model repository, as well as a questionnaire and more details on the annotation process.

### 5.1. Worst-case Accuracy versus Visual Analysis

We first confirm that visualization-based testing is more informative than global statistics. We show that the visualized performance (variance matrices) of four models can be very different even though they have the same (worst-case) accuracy score in Figure 8. The patterns of $\mathbf{M}_a$ suggest that $\mathbf{M}_a$ gives wrongly predicted testing images high confidence scores, and the outputs of the top-3 neurons triggered by foreground-only images are not consistent, which indicates that $\mathbf{M}_a$ might rely more on the background to make its decisions instead of the foreground objects. The abrupt green pattern at the top-left for $\mathbf{M}_c$ might be a signal of data leakage which needs further examination before being deployed. The yellow pattern on the edges of $\mathbf{M}_b$ suggests that $\mathbf{M}_b$ treats a few of testing images differently from the others. This could suggest that $\mathbf{M}_b$ might be sensitive to the presence of certain objects. For these reasons, our three annotators labelled $\mathbf{M}_a$ and $\mathbf{M}_c$ as *failed*, $\mathbf{M}_b$ as *borderline*, and $\mathbf{M}_d$ as *passed*. These four models have the same statistical evaluation scores (worst-case accuracy) whilst their visualized testing results can yield different judgements on their invariance qualities. To consolidate if we can use such visualization-based testing framework for background invariance testing, we evaluate 1) how diverse the testing

Table 1. Consistency, Reliability and Comprehensiveness Level of Different Background Invariance Testing Approaches

| | Random Sampling | Distance-based Sampling | | Keyword-based Sampling | |
|---|---|---|---|---|---|
| | | Nearest Top K | Interval (bin) | CLIP | Ontology (Ours) |
| Neuron Coverage (recall) [31] | 0.681 | 0.133 | 0.667 | 0.591 | 0.652 |
| Fleiss' reliability (precision) [13] | 0.384 | 0.906 | 0.531 | 0.640 | 0.649 |
| F1 score | 0.491 | 0.232 | 0.591 | 0.615 | **0.650** |

suites are, and 2) how consistent the human annotations are.

## 5.2. Comparisons Between Testing Methods

Human judgements of invariance qualities based on visualized testing results might vary across different testing runs. Therefore, we evaluate whether a testing method is desired based on: 1) diversity – neuron coverage rate (*recall*), and 2) consistency – inter-rater reliability (IRR) score (*precision*) as discussed in Section 4.3. Neuron coverage rates are often used to evaluate whether a selected testing suite is diverse enough to cover as many scenarios as possible [18, 31], and IRR scores are often used to evaluate the consistency and reliability of professional annotations [11] between different practitioners. We report f1 score (between diversity – recall, and reliability – precision) when comparing different testing methods.

### 5.2.1. Prior Works: Random/Nearest Sampling

In Table 1, we show that although the random sampling testing approach resulted in the best neuron coverage rate, the IRR score is relatively lower because of the randomness. Meanwhile, selecting the nearest top-$k$ background scenes resulted in the highest inter-rate reliability score, however, the neuron coverage rate (diversity level) is the lowest. Therefore, these two testing methods might not be the most suitable for background invariance testing.

### 5.2.2. Distance and Keyword-based Sampling

As shown in Table 1, although distance-based sampling (interval) can achieve a higher neuron coverage rate, the resultant human judgements are not most consistent, whereas keyword-based sampling methods are the most balanced between diversity and reliability. Among keyword-based sampling methods, CLIP [35] tends to find similar (matched) items for the target image, which leads to less diverse testing suites. Meanwhile, our ontology expands the originally detected keywords using association analysis and thus becomes the most balanced between diversity and consistency (i.e., the best f1 score). We also show the distribution of the distances between target images and the testing images synthesized using random, nearest neighbor, and our ontology-based method in Figure 11 in Appendix B5.

## 5.3. Automated Background Invariance Testing

To avoid the labor-intensive manual analysis process, we investigate if the entire testing procedure can be automated.

Table 2. Automation results: the automation accuracy using random forest is around 80% and the inter-rater reliability score with majority votes is around 0.65.

| | Automation Accuracy | IRR Score |
|---|---|---|
| Random Forest | $79.7 \pm 7.5\%$ | $0.649 \pm 0.091$ |
| AdaBoost | $74.8 \pm 9.1\%$ | $0.599 \pm 0.102$ |

We split the model repository into a training set (2/3 of the models) and a testing set (1/3 of the models). And we train a simple random forest to judge if models are rated as passed, failed or borderline using some hand-crafted features from the variance matrices [24]. To make the results more statistically significant, we randomly split the data, repeated the experiments ten times and reported the averaged results and the standard deviation. In Table 2, we show that we can achieve around 80% automation accuracy. Furthermore, the IRR scores between the predictions from assessors and the majority votes are similar to those of the three coders ($\sim$0.65). Therefore it shows the proposed framework can work as a fully automated background testing mechanism with sufficient accuracy.

# 6. Conclusion

In this work, we first confirm that visualization-based testing methods are more informative than reporting global statistics for background invariance testing. We show that models having the same averaged accuracy score can perform differently. With the proposed framework, we can visualize the testing results and find some visual patterns which facilitate further analysis of the invariance qualities.

We identify the challenges of utilizing visualization-based testing techniques when the data space is gigantic and adequate sampling is not feasible. We find that randomly sampled testing examples leads to inconsistent visualized patterns, hence inconsistent human decisions on invariance qualities across different testing runs. Meanwhile, nearest-neighbor sampling leads to consistent human judgment but with limited diversity in the sampled testing examples.

We propose an association ontology-based approach that can lead to 1) diverse testing suite, and 2) consistent and reliable human judgments on the invariance qualities of interested ML models. In the future, a larger model repository can be collected to confirm the feasibility of the framework.

## References


[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases (VLDB)*, pages 487–499, 1994. 4

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv:1907.02893*, 1, 2019. 2

[3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Transactions on Graphics*, 42(4), 2023. 5, 6

[4] Yu Cheng, Bo Yang, Bo Wang, and Robby T Tan. 3D human pose estimation using spatio-temporal networks with explicit occlusion training. In *Proc. AAAI Conference on Artificial Intelligence*, pages 10631–10638, 2020. 2

[5] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z Li, and Xudong Zou. Pedhunter: Occlusion robust pedestrian detector in crowded scenes. In *Proc. AAAI Conference on Artificial Intelligence*, pages 10639–10646, 2020. 2

[6] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *Proc. International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021. 2

[7] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (XAI): A survey. *arXiv preprint arXiv:2006.11371*, 2020. 3

[8] Jodi L Davenport and Mary C Potter. Scene consistency in object and background perception. *Psychological Science*, 15(8):559–564, 2004. 2

[9] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *Proc. 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 1, 2020. 13

[11] N El Dehaibi and EF MacDonald. Investigating inter-rater reliability of qualitative text annotations in machine learning datasets. In *Proc. Design Society: DESIGN Conference*, pages 21–30. Cambridge University Press, 2020. 7, 8

[12] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *Proc. International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019. 1, 3

[13] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378, 1971. 8

[14] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023. 5

[15] Xiang Gao, Ripon K Saha, Mukul R Prasad, and Abhik Roychoudhury. Fuzz testing based data augmentation to improve robustness of deep neural networks. In *Proc. IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 1147–1158. IEEE, 2020. 1

[16] Gösta Grahne and Jianfei Zhu. Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1347–1362, 2005. 4, 11

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 13

[18] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020. 6, 8

[19] Jinhan Kim, Robert Feldt, and Shin Yoo. Guiding deep learning system testing using surprise adequacy. In *Proc. IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 1039–1049. IEEE, 2019. 3, 6

[20] Tim Lauer, Tim HW Cornelissen, Dejan Draschkow, Verena Willenbockel, and Melissa L-H Võ. The role of scene summary statistics in object recognition. *Scientific Reports*, 8(1): 1–12, 2018. 2

[21] Jizhizi Li, Jing Zhang, Stephen J Maybank, and Dacheng Tao. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, 130:246–266, 2022. 7, 11

[22] Zukang Liao and Min Chen. Image similarity using an ensemble of context-sensitive models. In *Proc. 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1758–1769, 2024. 6

[23] Zukang Liao and Michael Cheung. Invariance testing and feature selection using sparse linear layers. In *Proc. 31st ACM International Conference on Information & Knowledge Management*, pages 4219–4223, 2022. 3

[24] Zukang Liao and Michael Cheung. Automated invariance testing for machine learning models using sparse linear layers. In *Proc. ICML: Workshop on Spurious Correlations, Invariance and Stability*, 2022. 8

[25] Zukang Liao, Pengfei Zhang, and Min Chen. ML4ML: Automated invariance testing for machine learning models. In *Proc. IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 34–41, 2022. 1, 3, 15, 16

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. 15

[27] Adrian Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, 2013. 15

[28] Augustus Odena, Catherine Olsson, David Andersen, and Ian Goodfellow. Tensorfuzz: Debugging neural networks with coverage-guided fuzzing. In *Proc. International Conference on Machine Learning*, pages 4901–4911. PMLR, 2019. 2, 6

[29] Sabine Öhlschläger and Melissa Le-Hoa Võ. Scegram: An image database for semantic and syntactic inconsistencies

in scenes. *Behavior Research Methods*, 49(5):1780–1791, 2017. 16

[30] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Transactions on Graphics*, 30(4):68, 2011. 6

[31] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *Proc. 26th Symposium on Operating Systems Principles*, pages 1–18, 2017. 2, 3, 6, 8, 12

[32] Javier Portilla and Eero P Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000. 2

[33] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv:1808.03305*, 1, 2018. 2, 16

[34] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv:1911.08731*, 1, 2019. 2

[35] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Proc. 36th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 8

[36] Noam Shoresh and Bang Wong. Data exploration, 2011. 1

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 1, 2014. 13

[38] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 1, 2022. 1

[39] Zhao Wang and Aron Culotta. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proc. AAAI Conference on Artificial Intelligence*, pages 14024–14031, 2021. 1

[40] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv:2006.09994*, 1, 2020. 1, 2, 7, 11, 12

[41] Xiaofei Xie, Lei Ma, Felix Juefei-Xu, Minhui Xue, Hongxu Chen, Yang Liu, Jianjun Zhao, Bo Li, Jianxiong Yin, and Simon See. Deephunter: a coverage-guided fuzz testing framework for deep neural networks. In *Proc. 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 146–157, 2019. 2, 6

[42] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI Conference on Artificial Intelligence*, pages 13001–13008, 2020. 2

[43] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2017. 11

[44] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 6, 11