Background Invariance Testing According to Semantic Proximity

Zukang Liao[1] **, Pengfei Zhang**[2] **and Min Chen**[1]

[1] University of Oxford, United Kingdom
[2] Tencent Video, AI R&D Centre, PCG, China
zukang.liao@eng.ox.ac.uk, joepfzhang@tencent.com, 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, intensity, and background invariance. Unlike many types of variance, the variants of background scenes cannot be ordered easily, which makes it difficult to analyze the robustness and biases of the models concerned. In this work, we present a technical solution for ordering background scenes according to their semantic proximity to a target image that contains a foreground object being tested. We make use of the results of object recognition as the semantic description of each image, and construct an ontology for storing knowledge about relationships among different objects using association analysis. This ontology enables (i) efficient and meaningful search for background scenes of different semantic distances to a target image, (ii) quantitative control of the distribution and sparsity of the sampled background scenes, and (iii) quality assurance using visual representations of invariance testing results (referred to as *variance matrices*). In this paper, we also report the training of an ML4ML assessor to evaluate the invariance quality of ML models automatically.

1. Introduction

There are a variety of invariance qualities associated with machine learned (ML) models. Testing invariance qualities enable 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. The testing also allows us to observe the possible biases or spurious correlations that may have been learned by a model (Wang and Culotta 2021) and to anticipate if the model can be deployed in other application domains (Wang et al. 2022). This work is concerned with background invariance testing – a relatively challenging type of testing.

Many types of commonly-deployed invariance testing focus on variables that can be ordered easily, such as sizes and rotation angles As illustrated by (Anonymised Authors 2022), ordering the testing results is important for observing the level of robustness in relation to the likelihood of different variations (e.g., a slightly rotated car vs. an upside-down car in an image). However, in background invariance testing, the term "background" is a multivariate variable and commonly expressed qualitatively (e.g., an outdoor scene, in a desert, and so on), making it difficult to order the testing results. Therefore, one cannot judge if a model is robust against certain variations, or how different background scenes influence spurious correlations. Furthermore, without a mechanism for ordering background scenes consistently, the ML4ML approach for automated invariance testing (Anonymised Authors 2022) cannot be used.

In the literature, some previous work focused on the quality of background scenes, e.g., introducing black pixels or random noise (Lauer et al. 2018) into the background. While this allows the variations to be ordered, the variations of image quality are indeed very different from the variations of background scenes. Other previous work focused on testing foreground objects against random background images (Xiao et al. 2020). While this approach can provide an overall statistical indication of the invariance quality, the background images are randomly selected, and it thus does 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 different background scenes.

In this paper, we present a technical solution to the need for ordering background scenes by utilizing semantic information. Our technical solution is built on the existing techniques of scene understanding in computer vision and those of ontological networks that are used in many text analysis applications. With this technique, we are able to:

a. Search for n different background scenes in a meaningful and efficient way based on the semantics encoded in each original image with a foreground object $\mathbf{x}$;

b. Control the distribution and sparsity of the n background scenes according to their semantic distance to the original image containing $\mathbf{x}$;

c. Construct n testing images from n background scenes for each $\mathbf{x}$ and test an ML model with the testing images;

d. Apply steps (a-c) to a large number of l target images, $\mathbf{x}_1, \ldots, \mathbf{x}_l$, and generate $l \times n$ testing images.

e. Test an ML model for object classification with the $l \times n$ testing images, collect results or intermediate results at k positions of the model, and transform the results to k visual representations (referred to as *variance matrices*) in a consistently-ordered manner.

f. Apply step (e) to a model repository of m different ML models, and use the resultant $k \times m$ variance matrices to train an ML4ML assessor for evaluating the invariance quality of ML models. With a trained ML4ML assessor, the process of background invariance testing can be automated since steps (a-e) can easily be automated.

2. Related Work

The invariance qualities of ML models have been studied for a few decades. In recent years, invariance testing becomes a common procedure in invariant learning (Arjovsky et al. 2019; Sagawa et al. 2019; Creager, Jacobsen, and Zemel 2021). 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.

(Rosenfeld, Zemel, and Tsotsos 2018) 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, (Zhong et al. 2020; Cheng et al. 2020; Chi et al. 2020) replaced parts of the images with black or grey pixels for foreground invariance testing.

(Davenport and Potter 2004) noticed that the association between a foreground object and its background scene affected object recognition and described such association as "consistency". (Lauer et al. 2018) 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 (Portilla and Simoncelli 2000).

Several researchers experimented with swapping background scenes in studying background invariance, e.g., (Davenport and Potter 2004). (Xiao et al. 2020) 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.

Like many invariance qualities (e.g., rotation, size, and intensity), it is relatively easy to control the variation of noise level, the size of the replacement patch, and the inconsistency level of color textures. However, it is not so easy to control the level of consistency or semantic association when one replaces one background scene with another. This work aims to address this research challenge.

Measuring semantic association between background scenes can benefit from existing scene understanding models, e.g., (Pan et al. 2018; Aditya et al. 2015). We refer interested readers to a few comprehensive surveys on scene understanding, including (Naseer, Khan, and Porikli 2018; Grant and Flynn 2017). In this work, we use two models, which were pre-trained on the ADE20k database (Zhou et al. 2017b) and the Place365 database (Zhou et al. 2017a) respectively, to extract semantic information from images.

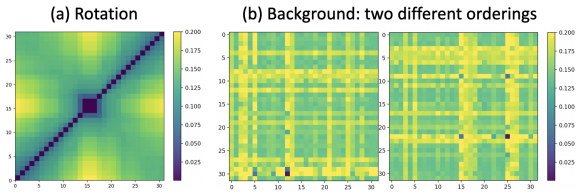

Figure 1: (a) The variance matrix obtained for rotation invariance testing is naturally ordered and its visual patterns are meaningful. (b) If the background scenes are not ordered consistently, the same set of background scenes may result in variance matrices with arbitrary visual patterns, which cannot be used to train ML4ML assessors.

Our solution also utilizes the techniques developed in other branches of AI and ML, including ontology (de Sousa Ribeiro and Leite 2021; Panigutti, Perotti, and Pedreschi 2020) and association analysis (Agrawal, Srikant et al. 1994).

3. Definition, Overview, and Motivation

Let $\mathbf{x}_i$ be the i^{th} image in a dataset D and o_i be the foreground object in $\mathbf{x}_i$. $\mathbf{M}$ be an ML model trained to recognise 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 set of testing images of $\mathbf{y}_{i,j}$. As reported by (Anonymised Authors 2022), when the testing results are organised into a variance matrix (Figure 1a), the visual patterns in the variance matrix can be analyzed using an ML4ML assessor to evaluate the invariance quality of $\mathbf{M}$.

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:

$$\mathbf{y}_{i,j} = B(o_i, \mathbf{b}_j),\ j = 1, 2, \ldots, n \quad (1)$$

generate n testing images. Similar to other invariance testing, the testing results can be summarized as a variance matrix. However, as these background scenes may be ordered according to their locations in a database, different orderings may yield different variance matrices (Figure 1b). The visual patterns in such a variance matrix are not as meaningful as those resulting from rotation transformations (Figure 1a).

If we can find a way to consistently produce variance matrices for background transformations, we can adopt the ML4ML invariance testing framework proposed by (Anonymised Authors 2022) for background invariance testing. This motivates us to address the following challenges:

1. We need to introduce a metric for measuring the semantic distance between each background scene and the corresponding target image $\mathbf{x}_i$.
2. We need to produce a variance matrix as a uniform data representation from non-uniform sampling of background scenes, as sampling background scenes will not

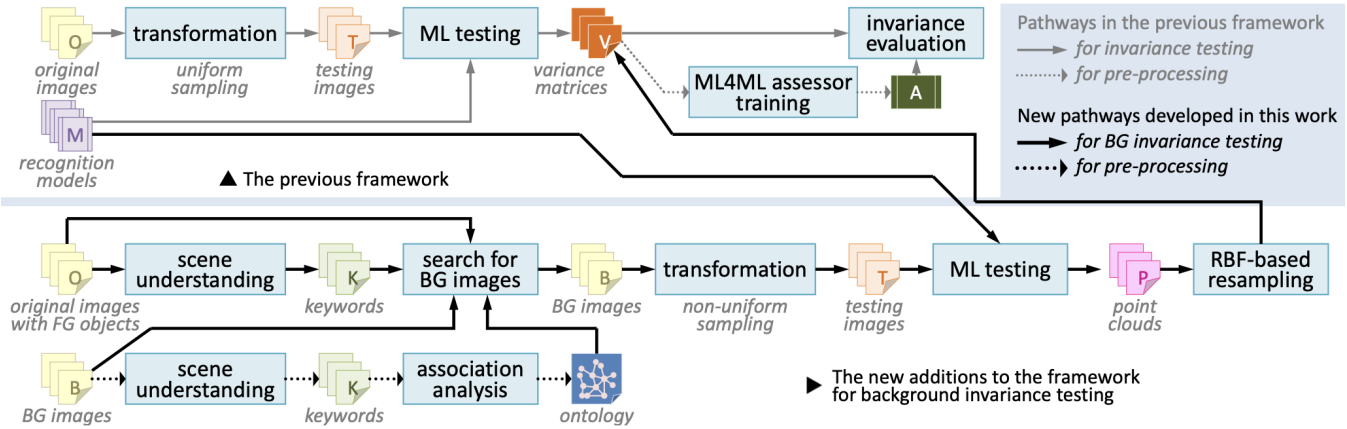

Figure 2: The upper part of the figure shows the previous framework proposed by (Anonymised Authors 2022), where the transformations for invariance testing are uniformly sampled. As the transformations for background invariance testing cannot easily be sampled uniformly, we propose to introduce a new sub-workflow (lower part) with an additional set of technical components for enabling non-uniform sampling of such transformations. This sub-workflow is detailed in Section 4. Methodology.

be as uniform as sampling sizes, rotation angles, and many other types of variables in invariance testing.

3. We need to have an effective way to search for background scenes that will be distributed appropriately for producing variance matrices.

As illustrated in Figure 2, the upper part of the figure shows the previous framework (Anonymised Authors 2022) for invariance testing involving uniform sampling of transformations. In this work, we introduce a number of new technical components (the lower part of the figure) to address the aforementioned challenges. Once these challenges are addressed, background invariance testing can be integrated into the ML4ML invariance testing framework.

4. Methodology

In this section, we follow the pathways in the lower part of Figure 2 to describe a series of technical solutions for enabling background invariance testing with non-uniform sampling of the transformations of the original images.

Consider a large collection of l original images $\mathbf{x}_i$, $i = 1, 2, \ldots, l$ to be tested and a large number of background scenes $\mathbf{b}_j$, $j = 1, 2, \ldots$ in an image repository $\mathbb{B}$. The first part of the sub-workflow is to identify a set of background scenes suitable for transforming each original image $\mathbf{x}_i$ (also called *target image*) with a specific foreground object o_i. The ML models to be tested for background invariance quality are expected to recognize o_i when o_i is combined with different background scenes, or to classify such combined images with the label of o_i.

Naturally, one may consider to use conventional image similarity metrics (e.g., cosine/l2 similarity used for metric learning (Kaya and Bilge 2019)) to find background scenes similar to $\mathbf{x}_i$. However, image similarity does not necessarily imply plausibility. Furthermore, as a suitable background scene may not (often is desirable not to) have the foreground object o_i, the similarity metrics cannot deal with the conflicting requirements, similar background and different foreground, easily. We therefore focus on the semantic distance between images (Challenge (1) in the previous section).

Image Semantics from Scene Understanding. Research on scene understanding aims to extract different semantic information from images. In this work, we represent the semantics of each image with the keywords extracted by employing existing scene understanding techniques to process the original images to be tested and background scenes to be used for transformation. From each image, $\mathbf{a}$ ($\mathbf{a} = \mathbf{x}_i$ or $\mathbf{b}_j$), a scene understanding model identifies a set of objects that are recorded as a set K_a of keywords. We detail the scene understanding model used in this work in Appendix A. When $\mathbf{a}$ is a target image, i.e., $\mathbf{a} = \mathbf{x}_i$, we assume that K_a contains a keyword for the foreground object o_i.

Figure 3 shows two examples of foreground images and two examples of background scenes. For some images, scene understanding may result in many keywords, but in other cases, only 1-3 keywords (e.g., the 1^{st} and 3^{rd} images in Figure 3). Therefore it is desirable to consider not only the keywords extracted from each image, but also the keywords related to the extracted keywords.

Ontology from Association Analysis. Ontology is a graph-based knowledge representation, which is used to store the relationships among different keywords in this work. As illustrated in Figure 4, nodes represent keywords, and an edge between two nodes indicates that two keywords have been extracted from the same image. 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 the repository $\mathbb{B}$.

The Apriori algorithm (Agrawal, Srikant et al. 1994) is widely used for association analysis. When the size of the dataset is great, the Frequent Pattern Growth algorithm (Grahne and Zhu 2005) can run more efficiently. Considering a set of all possible keywords K_{all} that can be extracted from all images in the repository $\mathbb{B}$, the level of association

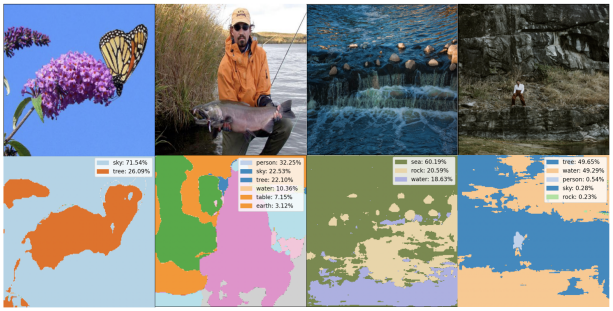

Figure 3: (a), (b) are target images. (c), (d) are background candidates. (a) and (c) have only a few detected keywords.

between two keywords k_a and k_b can be described by two concepts. Consider three itemsets: $s_a = \{k_a\}$, $s_b = \{k_b\}$, and $s_{ab} = \{k_a, k_b\}$, The first concept, *support* for the itemset s_{ab}:

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

indicates the frequency of the co-occurrence 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 the second concept *confidence*:

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

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 a large image repository, 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, and the improvement of scene understanding techniques. Hence, it is difficult to use the support values consistently. We therefore use the confidence values for weights on the directed edges in the ontology.

In the ontology, the shortest path between two keywords indicates the level of association between two keywords, typically facilitating two measures, (i) the number of edges along the path (i.e., hops) and (ii) an aggregated weight, e.g., $\prod_i w_{i=1}$ or $\min(0, w_1 - \sum_{i=2}(1 - w_i)^{\alpha_i}(\alpha_i \geq 1))$.

Background Scenes from Semantic Search. Given a target image $\mathbf{x}$, to test if an ML model is background-invariant, we would like to find a set of background scenes that can be used to replace the background in $\mathbf{x}$ while maintaining the foreground object o. The set of keywords K_x extracted by the 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, semantic search can work very well. However, as exemplified in Figure 4(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 (i.e., challenge (3) in Section 3), 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 4, 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:

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

Figure 5 shows three sets of example background scenes discovered for a targeting image (i.e., the fish image on the top-left corner of each set). The background scenes in the first set (left) are discovered by searching the image repository randomly. Those in the last set (right) are discovered using the initial set of keywords K_x. Those in the second set (middle) are discovered using an expanded set of keywords, OL_4, after the 4th expansion. While it is not necessary for every testing image in an invariance testing to be realistic, the plausibility of a testing image reflects its probability of being captured in the real world. As discussed in (Anonymised Authors 2022), it is unavoidable that invariance testing involves testing images of different plausibility, and therefore it is important to convey and evaluate the testing results with the information of the plausibility. An ideal set of background scenes should have a balanced distribution of scenes of different plausibility. Qualitatively, we can observe that in Figure 5, the random set has too many highly implausible images and the closest set has images biased towards keywords $K_x = \{$painting, water, tree$\}$, many are not quite plausible, while the expanded set has a better balance between more plausible to less plausible background scenes. In Appendix B, we measure plausibility quantitatively using semantic distance. And we show more details on the keyword expansion using the ontology and candidate selection process in Appendix A.

Testing Images from Background Replacement. The transformation process for sampling the background variations is more complex than other commonly-examined invariance qualities, requiring the use of a segmentation tool to separate the foreground object o_i from each target image $\mathbf{x}_i$, and then superimpose o_i into individual background scenes discovered in the previous step. As defined in Eq. 1, for n different background scenes $\mathbf{b}_1, \mathbf{b}_1, \ldots, \mathbf{b}_n$, the transformation process produces n testing images $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,n}$

When a background scene $\mathbf{b}_j$ also contains one or more objects of the same class label as the target object o_i, it creates two problems. (i) If o_i is superimposed onto $\mathbf{b}_j$ without removing those similar objects, this undermines the validity of the ML testing because when an ML model returns a label of o_i for $\mathbf{y}_{i,j}$, it is unknown that the model has recognized the superimposed o_i or the similar objects in $\mathbf{b}_j$. (ii) If those similar objects are removed, the resulting image $\mathbf{y}_{i,j}$ would have holes that may not be covered by the superimposed o_i.

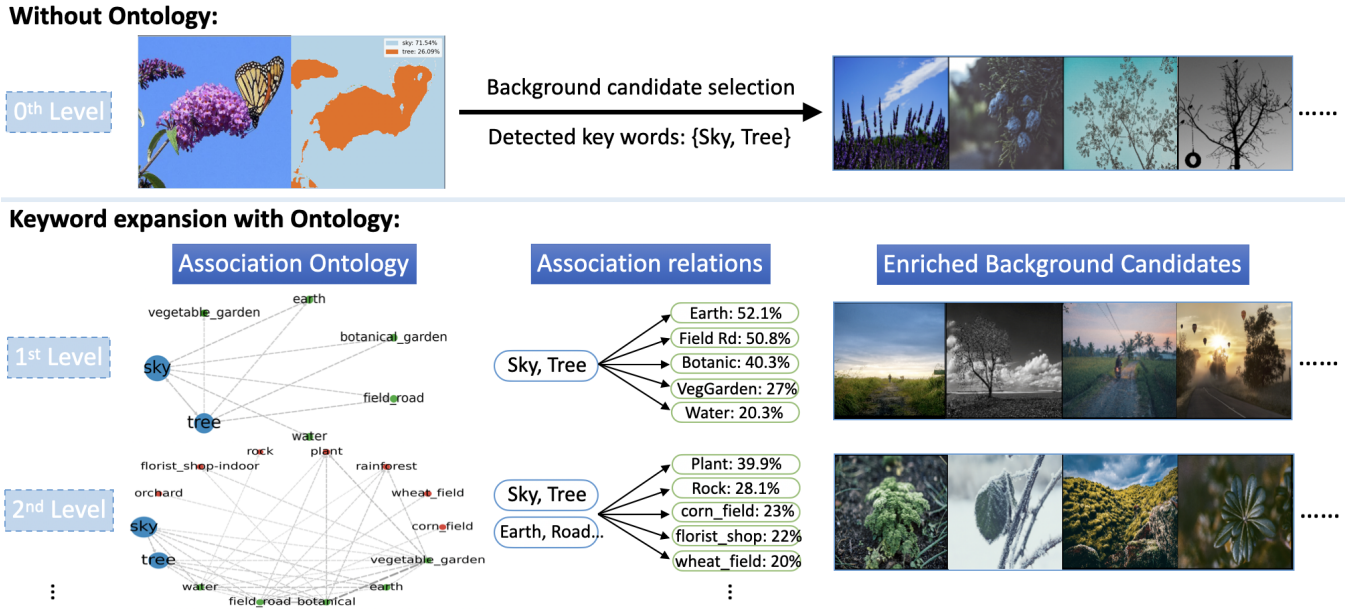

Figure 4: An example image has only two keywords detected by a pre-trained scene understanding model, namely $\{sky, tree\}$. Using an ontology, more keywords can be discovered iteratively, increasing the number and diversity of background scenes.

For these two reasons, we filter out any background scenes with the same keyword as o_i.

Point Clouds from ML Testing. Unlike the previous framework where the transformation process samples variations uniformly (e.g., $0°, 1°, 2°, \ldots$ for rotation angles), the testing images $\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,n}$ are a set of non-uniform sampling points. When we test an ML model $\mathbf{M}$ against the testing images, we can measure the results and intermediate results of $\mathbf{M}$ in many different ways. In (Anonymised Authors 2022), the different measurements are controlled by the notions of *position* and *modality*. The position indicates where in $\mathbf{M}$ the signal may be captured, e.g., Max@CONF (the confidence vector of the final predictions) and Max@CONV-1 (the map of the last convolutional layer). The modality indicates what mathematical function is used to abstract the signal vector or map at a position to a numerical measure, e.g., Mean or Max. Hence, within the context of an ML model $\mathbf{M}$, a fixed position (ps) and a fixed modality (md) for each testing image $\mathbf{y}_{i,j}$, ML testing results in a numerical measure $S(\mathbf{M}, ps, md, \mathbf{y}_{i,j})$.

In addition, we can measure the *semantic distance* between each testing image $\mathbf{y}_{i,j}$ and the target image $\mathbf{x}_i$. As shown in Eq. 2, the confidence concept in association analysis is non-commutative. We therefore always use the semantic distance starting from $\mathbf{x}_i$. This assigns the value $m(\mathbf{y}_{i,j})$ with a position away from $\mathbf{x}_i$.

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

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

indicates the variation between the two testing results. As the variation corresponds to positions $d_{i,j}$ and $d_{i,k}$, this 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 $\mathbf{x}_i$. In Appendix B, we list details on the semantic distance $d_{i,j}$ and $d_{i,k}$ obtained using the ontology.

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. The first column in Figure 6 shows five examples of such point clouds. Because the number and the distribution of these points depend on the set of background scenes, we can observe that when the level of expansion OL[i] (see Eq. 3) increases, the sampling has more data points and better distribution.

Variance Matrices from RBF-based Resampling. Because the sampling of background transformation is not uniform (i.e., challenge (2) in Section 3), unlike the previous framework in Figure 2, we have to consider the options of training an ML4ML assessor with point clouds or converting point clouds to variance matrices. We select the latter option primarily because, in the previous framework, ML4ML assessors are trained with ML experts' annotation of invariance quality based on their observation of variance matrices. Replacing variance matrices with scatter plots in the annotation process would introduce an inconsistency in the framework in general and annotation in particular. In the short term, this would not be desirable, but in the longer term, one should not rule out the possibility of training ML4ML assessors with scatter plots.

We use the common approach of radial basis functions (RBFs) to transform a point cloud into a variance matrix. For each element e in a variance matrix, an RBF defines a circular area in 2D, facilitating the identification of all data points

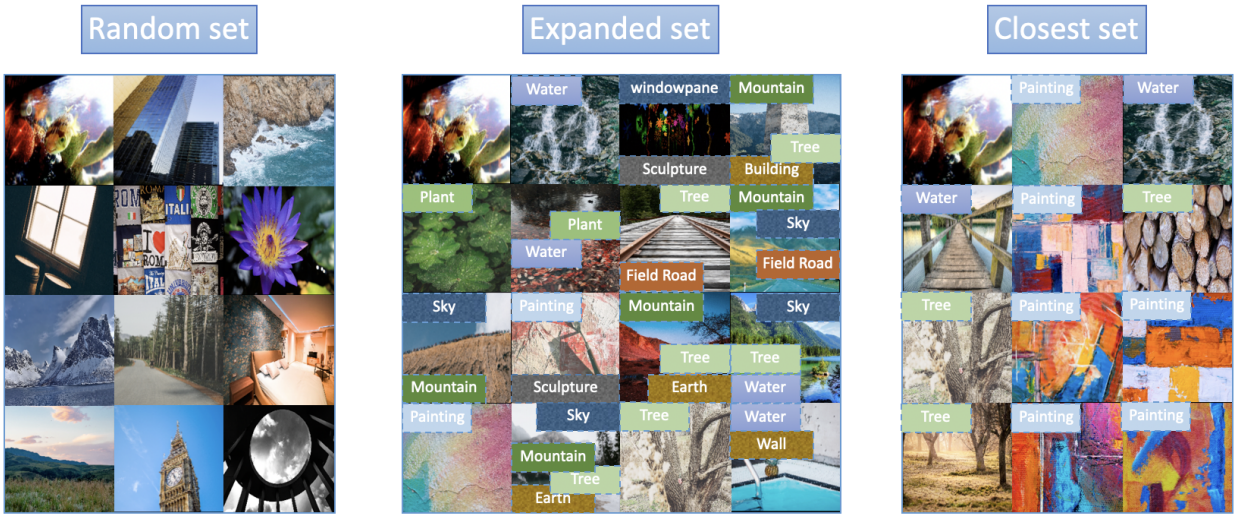

Figure 5: Three sets of example background scenes discovered for a target image of fish (the top-left of each set). The random set includes mostly unsuitable images. The closest set includes those discovered using only the original keywords $K_x = \{$painting, water, tree$\}$. The expanded set includes those discovered using the ontology, showing more suitable background scenes.

in the circle. Let these data points be $p_1, p_2, \ldots, p_c$ and their corresponding values are $v_1, v_2, \ldots, v_c$. As discussed earlier, the coordinates of each data point are determined by the semantic distances from the target image to two testing images. A Gaussian kernel ϕ is then applied to these data points, and produces an interpolated value for element e as

$$value(e) = \frac{\sum_{i=1}^{c} \big(\phi(\|e - p_i\|) \cdot v_k\big)}{\sum_{i=1}^{c} \phi(\|e - p_i\|)}$$

However, when the RBF has a large radius, the computation can be costly. When the radius is small, there can be cases of no point in a circle. In order to apply the same radius consistently, we define a new data point at each element e and use K nearest neighbors algorithm to obtain its value $u(e)$. The above interpolation function thus becomes:

$$value(e) = \frac{\phi(0) \cdot u(e) + \sum_{i=1}^{c} \big(\phi(\|e - p_i\|) \cdot v_k\big)}{\phi(0) + \sum_{i=1}^{c} \phi(\|e - p_i\|)}$$

In Figure 6, we show the application of three different RBFs. The mixed green and yellow patterns in row OL[1] gradually become more coherent towards OL[5]. We can clearly see a green square at the centre and yellow areas towards the top and right edges.

ML4ML Assessor Training and Deployment. With the variance matrices, we can use the same processes to train ML4ML assessors and deploy them to evaluate the background invariance quality of ML models in the same way as the previous framework. The same technical approaches in (Anonymised Authors 2022) can be adopted, including: (i) collecting variance matrices and ML models, (ii) splitting the model repository into a training and testing set and provide expert annotations of invariance quality based on variance matrices, (iii) engineering of imagery features for variance matrices, (iv) training ML4ML assessors using different ML techniques, and (v) testing and comparing ML4ML assessors.

5. Experiments

Testing Image Generation. We use the BG-20k (Li et al. 2022) database with 20,000 background images as all the candidates. We train a small repository of 250 models on the IN9 (smaller ImageNet) database (Xiao et al. 2020) to align with the previous attempts of background invariance testing (with randomly selected backgrounds).

For each model, we measure the signals at two positions, including the final predictions and the last convolutional layers (or the last layer before the final MLP head for Vision Transformers), as these two positions are considered important and interesting by our annotators. We use the $\max$ as the modality, and subtraction as the $\mathrm{dif}()$ operator in Eq. 4. As a result, for each model, we will have two original scatter plots (point clouds).

For RBF-based interpolation, we use the following parameters: (r=32, σ=10, K=32). For each model, we therefore obtain two variance matrices. For other settings of the interpolation, we refer interested readers to the Appendix B.

Training ML4ML Assessors. We build a small model repository of 250 models for object classification. The models were trained under different settings:

- Architectures: VGG13bn, VGG13, VGG11bn, VGG11 (Simonyan and Zisserman 2014), ResNet18 (He et al.

Table 1: IRR: Cohen's and Fleiss' kappa scores

Cohen's	Coder 1	Coder 2	Coder 3	Fleiss'
Coder 1	1	0.651	0.643	0.628
Coder 2	0.651	1	0.591	
Coder 3	0.643	0.591	1	

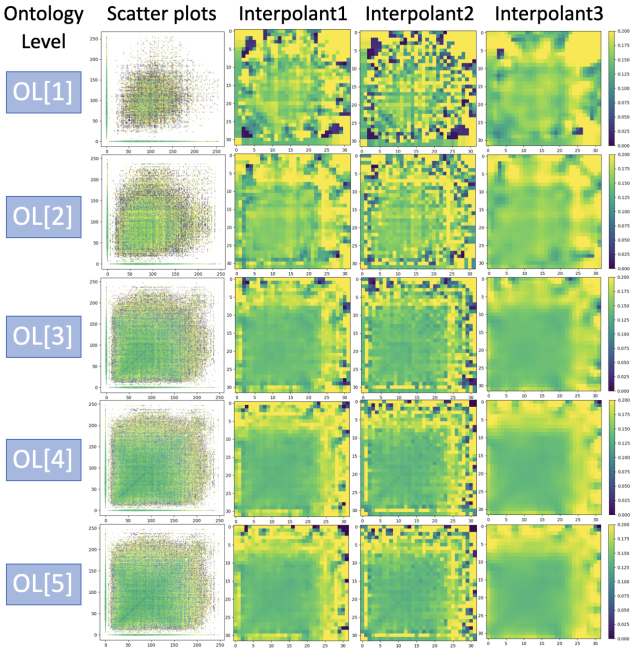

Figure 6: Original scatter plots (first column) and interpolated variance matrices (second to fourth column). The scatter plots are generated using the ontology level one to five, i.e., $OL[1], OL[2], \ldots, OL[5]$ for a randomly selected model M_{61}. Interpolant 1 - 3: (kernel size 16 σ=5), (kernel size 32 σ=2), and (kernel size 32 σ=10).

2016), and Vision Transformer (Dosovitskiy et al. 2020)

- Hyper-parameters: learning rate, batch size, epochs
- Augmentation: rotation, brightness, scaling, using images with only foreground (black pixels as background)
- Optimizers: SGD, Adam, RMSprop
- Loss: cross-entropy loss, triplet loss, adversarial loss

We use our sub-workflow in Figure 2 to generate testing images using the target images in the IN9 dataset, and the background scenes in the BG-20K database. We apply these models to the generated testing images. With the two measuring positions per model, we generate two variance matrices using the results from the ML Testing process.

For each model, we then provide professional annotations based on the original scatter plots and interpolated variance matrices 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 small questionnaire and more detail on the annotation process.

Finally, to automate the testing process, we extract the same set of statistical features as (Anonymised Authors 2022), e.g., mean / standard deviation, from the interpolated variance matrices. After that, we use the statistical features and the professional annotations of the training set of the model repository to train an ML4ML assessor, e.g., a random forest / adaboost in our case.

To evaluate the feasibility of the automation process using ML4ML assessors, we split the model repository into a training set (2/3 of the models) and a testing set (1/3 of the models). We do not tune any hyper-parameters for the ML4ML assessors, therefore we do not further split the training set into training and validation set. To make the results more statistically significant, we randomly split the data, repeat the experiments ten times and report the averaged results and the standard deviation.

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

	Automation Accuracy	IRR Score
Random Forest	79.7 + 7.5%	0.649+0.091
AdaBoost	74.8 + 9.1%	0.599+0.102
Worst-case Acc	64.4%	0.387

Results and Analysis In Table 2 we show that the majority votes of the professional annotations have around 0.4 IRR score with worst-case accuracy, which shows that the annotations are still aligned with the traditional accuracy metric. However, the professional labels are provided by considering the variance matrices and they are the decisions from many different factors instead of relying only on one single metric. Furthermore, Table 1 shows that the inter-rater reliability scores among the three professionals are around 0.65 which indicates that the annotations are consistent compared with many NLP tasks reported by (El Dehaibi and MacDonald 2020). Therefore, we believe that such annotations are both desirable and reliable.

In Table 2, we also show that by using ML4ML assessors (Anonymised Authors 2022), we could achieve around 70-80% automation accuracy which shows that the automation method (using ML4ML assessors) can achieve a satisfactory accuracy ($\sim$80%). Furthermore, the IRR scores between the predictions from ML4ML assessors and the majority votes are similar to those of the three coders ($\sim$0.6). Therefore it shows the proposed framework can work as a fully automated background testing mechanism with sufficient accuracy. For more studies and details on the ML4ML assessor, we refer interested readers to Appendix C.

6. Conclusion

In this work, we propose a technical solution to address a major limitation of an invariance testing framework recently reported by (Anonymised Authors 2022). The limitation is that background invariance testing cannot be incorporated into the framework as many other invariance problems. Our technical solution brings several non-trivial techniques together to overcome the three challenges in Section 3. The introduction of ontology is both novel and vital in making background invariance tests as meaningful as other commonly-seen invariance tests, e.g., rotation invariance. In this way, the previous framework has been expanded and improved significantly, paving the critical path for introducing other invariance tests with transformations that are not uniformly sampled, e.g., variations of clothing or hairstyles,

References

Aditya, S.; Yang, Y.; Baral, C.; Fermuller, C.; and Aloimonos, Y. 2015. Visual commonsense for scene understanding using perception, semantic parsing and reasoning. In *AAAI spring symposium series*.

Agrawal, R.; Srikant, R.; et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, 487–499. Citeseer.

Anonymised Authors. 2022. Anonymised Title. In *Proc. Anonymised Conference Name*. The paper is included in the supplementary materials.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv:1907.02893*.

Cheng, Y.; Yang, B.; Wang, B.; and Tan, R. T. 2020. 3d human pose estimation using spatio-temporal networks with explicit occlusion training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10631–10638.

Chi, C.; Zhang, S.; Xing, J.; Lei, Z.; Li, S. Z.; and Zou, X. 2020. Pedhunter: Occlusion robust pedestrian detector in crowded scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10639–10646.

Creager, E.; Jacobsen, J.-H.; and Zemel, R. 2021. Environment inference for invariant learning. In *International Conference on Machine Learning*, 2189–2200. PMLR.

Davenport, J. L.; and Potter, M. C. 2004. Scene consistency in object and background perception. *Psychological science*, 15(8): 559–564.

de Sousa Ribeiro, M.; and Leite, J. 2021. Aligning artificial neural networks and ontologies towards explainable ai. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 4932–4940.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

El Dehaibi, N.; and MacDonald, E. 2020. INVESTIGATING INTER-RATER RELIABILITY OF QUALITATIVE TEXT ANNOTATIONS IN MACHINE LEARNING DATASETS. In *Proceedings of the Design Society: DESIGN Conference*, volume 1, 21–30. Cambridge University Press.

Grahne, G.; and Zhu, J. 2005. Fast algorithms for frequent itemset mining using fp-trees. *IEEE transactions on knowledge and data engineering*, 17(10): 1347–1362.

Grant, J. M.; and Flynn, P. J. 2017. Crowd scene understanding from video: a survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(2): 1–23.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Kaya, M.; and Bilge, H. Ş. 2019. Deep metric learning: A survey. *Symmetry*, 11(9): 1066.

Lauer, T.; Cornelissen, T. H.; Draschkow, D.; Willenbockel, V.; and Võ, M. L.-H. 2018. The role of scene summary statistics in object recognition. *Scientific reports*, 8(1): 1–12.

Li, J.; Zhang, J.; Maybank, S. J.; and Tao, D. 2022. Bridging composite and real: towards end-to-end deep image matting. *International Journal of Computer Vision*, 130: 246–266.

Naseer, M.; Khan, S.; and Porikli, F. 2018. Indoor scene understanding in 2.5/3d for autonomous agents: A survey. *IEEE access*, 7: 1859–1887.

Pan, X.; Shi, J.; Luo, P.; Wang, X.; and Tang, X. 2018. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Panigutti, C.; Perotti, A.; and Pedreschi, D. 2020. Doctor XAI: an ontology-based approach to black-box sequential data classification explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 629–639.

Portilla, J.; and Simoncelli, E. P. 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, 40(1): 49–70.

Rosenfeld, A.; Zemel, R.; and Tsotsos, J. K. 2018. The elephant in the room. *arXiv:1808.03305*.

Sagawa, S.; Koh, P. W.; Hashimoto, T. B.; and Liang, P. 2019. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv:1911.08731*.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, Z.; and Culotta, A. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14024–14031.

Xiao, K.; Engstrom, L.; Ilyas, A.; and Madry, A. 2020. Noise or signal: The role of image backgrounds in object recognition. *arXiv:2006.09994*.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2020. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13001–13008.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017a. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40: 1452–1464.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017b. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.