ATOM: Automated Black-Box Testing of Multi-Label Image Classification Systems

Shengyou Hu†, Huayao Wu†*, Peng Wang†, Jing Chang‡, Yongjun Tu‡,
Xiu Jiang‡, Xintao Niu†, and Changhai Nie†
†State Key Laboratory for Novel Software Technology and
Department of Computer Science and Technology, Nanjing University, China
‡Guangdong OPPO Mobile Telecommunications Corp., Ltd., China
{husy, pengwang}@smail.nju.edu.cn, {hywu, niuxintao, changhainie}@nju.edu.cn
{changjing, tuyongjun1, jiangxiu}@oppo.com

***Abstract*—Multi-label Image Classification Systems (MICSs) developed based on Deep Neural Networks (DNNs) are extensively used in people's daily life. Currently, although there are a variety of approaches to test DNN-based systems, they typically rely on the internals of DNNs to design test cases, and do not take the core specification of MICS (i.e., correctly recognizing multiple objects in a given image) into account. In this paper, we propose ATOM, an automated and systematic black-box testing framework for testing MICS. Specifically, ATOM exploits the *label combination* as the testing adequacy criteria, hoping to systematically examine the impact of correlations between a fixed number of labels on the classification ability of MICS. Then, ATOM leverages image search engine and natural language processing to find test images that are not only common to the real-world, but also relevant to target label combinations. Finally, ATOM combines metamorphic testing and label information to realize test oracle identification, based on which the ability of MICS in classifying different label combinations is evaluated. To evaluate the effectiveness of ATOM, we have performed experiments on two popular datasets of MICS, *VOC* and *COCO* (each with five state-of-the-art DNN models), and one real-world photo tagging application from our industrial partner. The experimental results reveal that the performance of current DNN-based MICSs remains less satisfactory even in recognizing correlations between only two labels, as ATOM triggers a total number of 6,049 such label combination related errors for all MICSs studied. In particular, ATOM reports 587 error-revealing images for the industrial MICS, in which 92% of them are confirmed by the developers.**

***Index Terms*—Multi-label Image Classification Testing, Black-box Testing, Metamorphic Testing**

I. Introduction

Multi-label image classification is a fundamentally important task in computer vision, which plays an indispensable role in many application domains [1]–[3]. In contrast to single-label classification, multi-label image classification is more practical and challenging in the real-world, because a realistic image usually contains multiple objects and the correlations between these objects are difficult to capture. To implement an effective Multi-label Image Classification System (MICS), approaches based on Deep Neural Networks (DNNs) have been widely studied and achieved remarkable success [4]–[9]. However, unexpected behaviors of MICS are still frequently observed in practice [10], [11], which poses a pressing need to test such DNN-based MICSs thoroughly.

Currently, there are already many white-box approaches to test DNN-based systems [12]–[19]. These approaches typically generate test cases based on the internals of DNNs, such as network structure (e.g., neuron coverage and its variants [12]–[18]) or the training data (e.g., surprise adequacy [19]). However, in industrial test scenarios, such internal information is not always available to testers. For example, testers might be unable to access the code base or data set of the development team due to security or privacy concerns; they might also need to test systems that are provided by third parties. In this case, adopting white-box approaches is much more difficult, while the black-box approach becomes a more practical choice.

With the goal of black-box testing of DNN-based systems, some test coverage criteria like manifold combination coverage [20] and input diversity [21] are proposed. However, these criteria are mainly designed for testing general DNNs, which do not take the core functionality of MICS into account. In particular, modern DNN-based MICSs are typically developed to exploit the correlations between labels to improve classification accuracy [4]–[9], [22], and unexpected behaviors might thus occur if such correlations are incorrectly captured. For example, a buggy MICS might incorrectly capture the correlations between the *mouse* and the *keyboard*, so that it will always recognize the presence of a keyboard even if there is only a mouse in the image [23]. Clearly, such label correlations related errors should be carefully examined, but the large number of labels involved in modern MICSs will usually yield a potentially enormous label space that cannot be exhaustively tested.

In addition, when testing MICS, practical testers are generally more interested in realistic test images rather than synthetic ones (e.g., by directly putting an object on an existing image). Despite that testing approaches like fuzzing [24]–[27] and adversarial neural networks [12], [28] can effectively generate test images to trigger potential errors of MICS, these errors might not be well acknowledged by industrial developers just because the corresponding error-revealing images are not in a real scene (that is, unlikely to be encountered by actual users).

* Corresponding author

Meanwhile, to reduce manual efforts for checking the classification outputs of MICS, the test oracle identification should be automated. Current testing approaches of DNN-based systems either use differential testing [12], [17], [29]–[32] to determine whether the same test input will lead to different behaviors of multiple system implementations, or rely on metamorphic testing [26], [28], [33]–[38] to determine whether the system will exhibit specific behaviors when small perturbations are introduced to the test input. However, these approaches are often used to test either the correctness or the robustness of the system with respect to certain test inputs, but not to evaluate the capability of MICS in handling specific label combinations.

In this paper, we propose ATOM (Automated Black-Box Testing Of MICS), a novel black-box testing framework that automatically and systematically tests MICS. First, to establish a black-box test coverage criterion, we introduce the concept of *k-label combination* to measure the ability of MICS to handle the correlations between k labels. This criterion provides a systematic way to examine the numerous correlations inherent in the label space of MICS. Accordingly, for a given value of k, the set of all possible k-label combinations is exactly the candidate test targets that ATOM seeks to cover.

Next, to collect realistic test images for each k-label combination, ATOM leverages image search engine, because the Internet naturally provides a rich and diverse set of candidate images, and the volume and content of the images returned can also provide hints for ATOM to determine whether a k-label combination is indeed *common* to the real-world. Specifically, to reduce the influence of ambiguity of labels on the search accuracy, ATOM first relies on the tree structure representation of the MICS's label space and particular search operators to construct search keywords. Then, for the top-ranked images returned, ATOM further utilizes natural language processing to analyze the caption accompanying each image to remove images that are *irrelevant* to the k-label combination (i.e., the k objects are not present in the images). After this, if no image can be found, then ATOM considers the k-label combination as a *rare* label combination, because the images containing these k objects are unlikely to exist in the real-world (e.g., *airplane* and *frisbee*). Otherwise, the k-label combination is referred to as a *common* label combination, and the corresponding images collected will be used as the test inputs of MICS.

Finally, for the test oracle identification, ATOM combines metamorphic testing (with common image processing operations as metamorphic relations) and label information to determine the concrete classification ability of MICS on different common label combinations. For a test image of a common k-label combination, if it passes metamorphic testing, and all the k labels are also correctly recognized by MICS, then ATOM will conclude that the MICS has a reasonably good classification ability on this common label combination. While if the test image fails metamorphic testing, then there must be an error in MICS. However, such an error might not always be due to the recognition of this common k-label combination, but the influences of other unexpected objects in the test images. For this case, ATOM will use a heuristic strategy to analyze the outputs of both source and follow-up test images to determine whether the error observed is indeed relevant to the common k-label combination.

To evaluate the performance of ATOM, we have performed experiments on two popular datasets of MICS, *VOC* [39] and *COCO* [40], in which ten state-of-the-art DNN models are used (five DNNs for each dataset). From the experimental results, although all subject MICSs can achieve at least 81% mean average precision on their respective validation sets, we find that their performance remains less satisfactory even in recognizing correlations between only two labels. In particular, ATOM reveals a total number of 6,049 2-label combination related errors for all subject MICSs, and there are 70% and 24% of common 2-label combinations that are examined by ATOM but are not present in the validation sets of the two open-source datasets, *VOC* and *COCO*, respectively. Moreover, we have applied ATOM to test a real-world photo tagging application from our industrial partner, OPPO. A total number of 587 error-revealing images are reported, in which 92% of them are confirmed by the developers.

Overall, the key contributions of this paper are as follows:

- We introduce k-label combination, a black-box test coverage criterion, to measure the ability of MICS to handle correlations between k different objects.
- We leverage the image search engine and natural language processing to collect realistic test images that are relevant to each k-label combination.
- We propose to combine metamorphic testing and label information to determine the concrete classification ability of MICS on different label combinations.
- We incorporate the above inventions into a black-box testing framework, ATOM, and evaluate its effectiveness using two academic datasets and one real-world MICS product.

The remainder of this paper is organized as follows. Section II presents the proposed ATOM framework. Section III describes the research questions and experimental setup for evaluating ATOM. Section IV reports experimental results. Section V discusses the threats to validity. Section VI summarizes related work, and Section VII concludes this paper.

II. The ATOM Framework

This section presents the ATOM framework that tests MICS. Fig. 1 gives its overall workflow. Given a label space and a value of k indicating the label combination coverage, ATOM will first iterate every possible k-label combination, and rely on the image search engine to collect realistic and relevant test images. These test images will then be given to the MICS under test, and ATOM will combine metamorphic testing and label information to realize test oracle identification.

A. Label Combination

In order to test the core functionalities of MICS in a black-box manner, we propose to use the *label combination* as the test coverage criterion. For the multi-label image classification

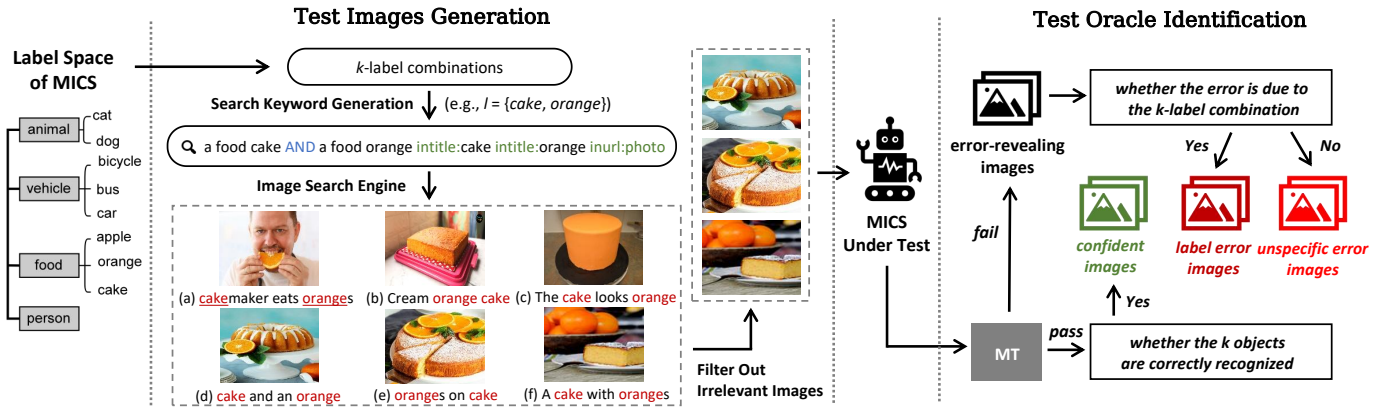


Fig. 1. The overall workflow of the ATOM framework.

task, each label c_i represents a single object that the MICS should identify. A set of d labels thus constitutes the label space, $\mathcal{Y} = \{c_1, c_2, \ldots, c_d\}$, of the MICS. For convenience, the fine-grained labels in the label space can be further grouped into several coarse-grained super categories when constructing the label space and dataset of the MICS [39], [40] (e.g., the labels *dog* and *cat* can be grouped into the *animal* super category). Accordingly, a tree structure is typically used to depict relationships between labels (leaf nodes) and their super categories (parent nodes) in modern MICSs [39], [40]. For example, Fig. 1 gives a label space that consists of nine labels (with four super categories). If the label has no super category (e.g., *person*), then the super category is the label itself.

In this study, we refer to a k-label combination, $l = \{o_1, \ldots, o_k\}$, $o_i \in \mathcal{Y}$ for $1 \leq i \leq k$, as a combination of k different labels in the label space $\mathcal{Y}$. The core idea is inspired by combinatorial testing [16], [41], [42], which is a popular testing technique to examine interactions of factors that influence the system's behavior. Here, each label combination exactly indicates a potential correlation of the k labels that a MICS should properly handle, and therefore, a test target that should be covered. Accordingly, by examining all possible k-label combinations, the core classification ability of MICS can be systematically tested.

B. Test Images Generation

Given the set of all possible k-label combinations L_k as the candidate test targets, ATOM will leverage image search engine (Google image search in this study) and natural language processing to collect test images that are not only common to the real-world, but also as relevant to these label combinations as possible. This process will produce at most N test images for each label combination ($N = 5$ in this study).

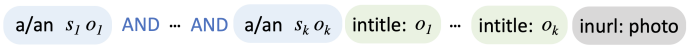


Fig. 2. The keyword used for label combination $l = \{o_1, \cdots, o_k\}$, where s_i indicates the super category of o_i.

1) Search Keyword Generation: For each label combination $l = \{o_1, \ldots, o_k\} \in L_k$, ATOM follows the pattern shown in Fig. 2 to construct the search keyword. First, ATOM uses `AND` to concatenate all k labels to form the initial keyword. Then, due to the linguistic ambiguity in some labels (e.g., the label *mouse* can indicate either an animal, or a computer device), ATOM adds the super category s_i of each label o_i into the keyword to reduce the impact of ambiguity on the search accuracy (as the super category could provide additional semantic information of the labels). Meanwhile, the search operators `inurl` and `intitle` are also added to improve the search accuracy. The use of `inurl:photo` could help to narrow down the search results to websites containing realistic images (rather than cartoons, icons, or screenshots); and the use of `intitle:`o_i could help to increase the chance that the images returned actually contain the target label o_i.

In addition, because the same object can be expressed using different words in the natural language (e.g., *bicycle* and *bike*), ATOM further relies on a synonym directory to expand the set of candidate keywords. Specifically, for each label $o_i \in l$, we use *WordNet*[1], a large lexical database of English, to automatically construct its synonym set SYN_i (we let o_i be an element in SYN_i). These sets obtained are then manually checked and refined to form the final synonym directory. On this basis, for each k-label combination, all choices in the Cartesian product of the k synonym sets, $SYN_1 \times \cdots \times SYN_k$, will be used by ATOM to construct the keywords (the keyword consisting of initial labels is called the *default* keyword).

For example, for the label combination $l = \{cake, bicycle\}$ in Fig. 1, we have $o_1 = cake$ (with super category $s_1 = food$) and $o_2 = bicycle$ (with $s_2 = vehicle$). Assuming that the synonym sets of these two labels are as $SYN_1 = \{cake\}$ and $SYN_2 = \{bicycle, bike\}$. Then, we have $SYN_1 \times SYN_2 = \{(cake, bicycle), (cake, bike)\}$. Accordingly, ATOM will construct the following two keywords: *a food cake AND a vehicle bicycle intitle:cake intitle:bicycle inurl:photo* and *a food cake AND a vehicle bike intitle:cake intitle:bike inurl:photo*, where the former one is the default keyword.

[1]https://wordnet.princeton.edu/

2) Filtering Out Irrelevant Images: For each search keyword, ATOM acquires the top-M images returned from the image search engine ($M = 20$ in this study). ATOM then uses *Spacy*[2], a natural language processing library, to analyze the caption accompanying each of these images, in order to filter out images that are potentially irrelevant to the corresponding k-label combination (i.e., the image does not contain the k specific objects). Specifically, for each image caption, ATOM first applies lemmatization to transfer all words involved into their respective base forms, and checks whether every label occurs in the caption. If any label does not occur (the object is unlikely in the image), or any two labels occur consecutively (the two labels tend to describe a single object in the image), then the image will be removed. Further, ATOM applies part-of-speech tagging to determine the part-of-speech of each word in the caption. If the label occurs in the caption but its part-of-speech is not *noun* (the label tends to be used to describe another word), then the image will also be removed.

For example, for the six images (and their captions) shown in Fig. 1, images (a), (b), and (c) will be removed. For image (a), *cakemaker* but not *cake* appears in the caption; for image (b), *orange* and *cake* appear in succession in the caption; for image (c), *orange* is in the adjective form in the caption.

3) Overall Search Process: With the set of candidate search keywords and filtering strategy discussed above, ATOM seeks to iterate every search keyword for each k-label combination (starting from the default keyword), until a number of N relevant test images are collected. After this, if there remain no N images, ATOM tries to use the keyword of the simplest form, o_1 (s_1) `AND` $\cdots$ `AND` o_k (s_k), to query the image search engine, hoping to find any potentially relevant images.

After the search process, if at least one relevant test image can be collected for a k-label combination, then this label combination is referred to as a *common* k-label combination (that is, the correlation between the k labels is common to the real-world, and will thus be a test target of ATOM). The remaining k-label combinations are considered *rare*, as there tends to have a low chance that the k objects can appear together in a realistic image (accordingly, their correct classifications are generally not required).

C. Test Oracle Identification

Once the set of common k-label combinations, $L \subseteq L_k$, and their corresponding test images are obtained, ATOM will pass these test images to the MICS to get the classification outputs. Then, ATOM will identify the test oracle, and evaluate the concrete classification ability of the MICS on each $l \in L$.

Algorithm 1 gives the detailed oracle identification process of ATOM. For each common k-label combination $l \in L$ and its corresponding test images X_l, ATOM seeks to classify each of these images $x \in X_l$ into three groups (Lines 4–14): images that can be correctly handled by the MICS (*confident images*, C), images that reveal the label combination l related errors (*label error images*, E), and images that reveal errors that are not due to l (*unspecific error images*, U). To this end, ATOM first applies metamorphic testing to determine whether there is an error (by checking whether the MICS is robust to small perturbations introduced to the test images), and then utilizes label information to determine whether the error observed is indeed related to the common k-label combination l.

Algorithm 1 The Oracle Identification Process of ATOM

Input: MICS f, a set of common label combinations L, test images X_l for each $l \in L$, and a set of metamorphic relations R_i $(1 \leq i \leq m)$
Output: Sets of confident images C, label error images E, unspecific error images U, confident label combinations L_c, and vulnerable label combinations L_e

1: $C, E, U, L_c, L_e \leftarrow \emptyset$
2: **for** $l \in L$ **do**
3: **for** $x \in X_l$ **do**
4: $x_i \leftarrow R_i(x)$ for $1 \leq i \leq m$ ▷ follow-up images
5: $x_0 = x$
6: **if** x does not violate any MR **then**
7: **if** $l \subseteq f(x)$ **then**
8: add x into C ▷ confident image
9: add l into L_c
10: **else** ▷ fail MT
11: **if** $l \subseteq \cup_{i=0}^{m} f(x_i)$ **and** $l \not\subseteq \cap_{i=0}^{m} f(x_i)$ **then**
12: add x into E ▷ label error image
13: add l into L_e
14: **else**
15: add x into U ▷ unspecific error image
16: **return** C, E, U, L_c, L_e

Specifically, for each test image x (i.e., source test image), ATOM first adopts a given set of m metamorphic relations (MRs) to generate m follow-up test images, $\{x_1, \ldots, x_m\}$. Here, if x does not violate any MR, i.e., $f(x) == f(x_i)$ for $1 \leq i \leq m$, ATOM will further check whether $l \subseteq f(x)$ is satisfied to determine whether the common k-label combination l is correctly classified (i.e., whether all labels in l appear in the classification output). If this is the case, x is considered as a *confident image* of l, and l is considered as a *confident label combination* (Lines 8–9), indicating that the MICS has a reasonably good classification ability on l. Note that ATOM relies on the subset, rather than the equivalence, relation to determine the correctness of classification. This is because the goal of ATOM is to test the classification ability of MICS on each specific k-label combination l at each time (i.e., to ensure that the MICS does not ignore or misclassify any of the k objects in l). While whether the MICS is able to correctly recognize objects other than those involved in l will be examined when testing other label combinations.

If the test image x violates any MR (Lines 11–14), then there must be an error, and the next step is to determine whether the error is related to the common k-label combination l. To this end, ATOM applies a heuristic strategy to compare l against all classification outputs observed. First, ATOM checks

[2]https://spacy.io/

Table I: The Metamorphic Relations and Their Parameter Values

MR	Operation	Parameter	Parameter Value
R_1	Scale	Scalar	0.8
R_2	Brightness	Brightness factor	0.8
R_3	Contrast	Contrast factor	0.8
R_4	Rotation	Rotation angle	2°
R_5	Gaussian Blur	Radius	1
R_6	Sharpness	Sharpness factor	0.8
R_7	Saturation	Saturation factor	0.8

whether l appears in the union of outputs of both source and follow-up images, i.e., whether $l \subseteq \cup_{i=0}^{m} f(x_i)$ is satisfied (for naming consistency, we refer to the source image as x_0). If so, then the source image is highly likely to contain all objects in l and is a valid test input (as the object that is a bit hard to recognize in the source image might be recognized when some small perturbations are introduced). Next, ATOM checks whether l does not appear in the intersection of all outputs, i.e., whether $l \not\subseteq \cap_{i=0}^{m} f(x_i)$ is satisfied. If so, then the violation of MR is highly likely due to the misclassification of l (if all outputs contain l, then the MICS is likely to recognize l, and the error is thus due to other unexpected objects in the image). Finally, if both of the above two conditions are satisfied, then x is considered as a *label error image* of l, and l is considered as a *vulnerable label combination* (Lines 12–13). Otherwise, x is considered as an *unspecific error image* (Line 15), in which the error might be due to the misclassification of other label combinations except for l.

In this study, we select $m = 7$ image processing operations that are common to real-world photo applications as the MRs to implement Algorithm 1. Table I summarizes these MRs and their parameter values used. Specifically, the first five MRs (from R_1 to R_5) indicate widely used MRs in previous studies of testing image recognition systems [28], [43]. We note that although the *Rotation* can lead to changes in each pixel in the source image, the objects in the image typically remain intact and can be easily recognized by humans. In particular, the design of some modern DNNs has already taken this scenario into account, in which the rotation operation is used as a data augmentation strategy to enhance the training data [44], [45]. So, it is reasonable to examine whether the modern DNN-based MICSs could properly handle rotated images with small rotation angles. In addition, to account for the test scenario of our industrial partner (testing a photo tagging application), we further add two more operations as the MRs, i.e., R_6 and R_7. We note that these two operations are also widely used data augmentation strategies in practice [44].

With respect to the parameter values of these MRs, we find that different studies usually use different settings (e.g., for *Contrast*, some studies use the range between 1.2 and 3.0 [28], while some others use $[0.5, 2]$ [46]). In this study, we choose to set these parameter values in a more conservative manner (i.e., use relatively small values), hoping to introduce as fewer perturbations to the source test images as possible (as such, the semantics of test images are more likely to be preserved). The parameter values of the two newly added MRs were determined with the same goal in mind. We have performed preliminary experiments on 100 randomly selected test images of our experiments, and manually verified that the parameter values used will not lead to a change in human recognition.

Table II: The Datasets and DNN Models Used in the Experiments

Dataset	Size of Label Space	MICS	mAP
VOC [39]	20	MSRN [47]	96.0%
		MLGCN [48]	94.0%
		DSDL [49]	94.4%
		ASL [50]	94.6%
		MCAR [51]	94.8%
COCO [40]	80	MSRN [47]	83.4%
		MLGCN [48]	83.0%
		DSDL [49]	81.7%
		ASL [50]	86.6%
		MLD [52]	90.0%
PHOTO	32	-	-

III. Experimental Setup

This section presents the experiments we performed to evaluate the performance of ATOM. In particular, we set up the following three research questions:

- **RQ$_1$** How effective is ATOM in collecting realistic and relevant test images for testing MICS?
- **RQ$_2$** How effective is ATOM in revealing potential errors in MICS?
- **RQ$_3$** How effective is current MICS in handling correlations between different labels?

A. Systems Under Test

In this study, we selected two popular datasets of multi-label image classification, *VOC* [39] and *COCO* [40], as the experiment subjects (*VOC* contains 9,963 images of 20 labels; *COCO* has a larger scale, which contains 122,218 images of 80 labels). For each of these two datasets, we selected five state-of-the-art DNN models as the MICS under test. Table II gives the names of these DNNs used, and the mean average precision (*mAP*) that can be achieved on the validation set of the datasets. Here, all of these DNNs are pre-trained on the training set of *ImageNet* [53], except for *MLD* [52] of *COCO*, which is pre-trained on *Open Images* [54]. We used the code provided by the authors of [51] to fine-tune the *MCAR* [51] on *VOC*, as this DNN is not publicly available. Apart from this, the other nine DNNs (which have been fine-tuned on *VOC* and *COCO*) are directly obtained from their respective GitHub repositories [55]–[60].

In addition to the above DNNs from academia, we further included a real-world DNN-based photo tagging application, *PHOTO*, in our experiments. This MICS comes from our industrial partner, OPPO, which is released as a component of the photo app of a popular mobile operating system (which has over 500 million active users per month). As discussed

in Section I, the testing team of this MICS cannot access the internals of the DNN model, so black-box testing is the only feasible choice. Since the label space of this MICS remains evolving, we asked the testing team to identify the most important labels and used the 32 labels received to form its label space. Due to the confidentiality agreement, we cannot disclose the concrete label space, the name of the DNN model used, and the *mAP* of this MICS.

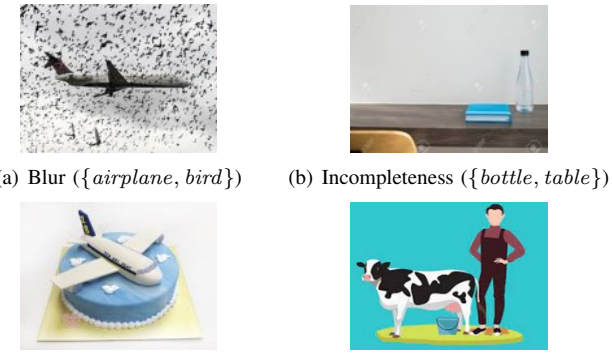
B. Experiment Process

In this study, we applied ATOM to examine k-label combinations for $k = 1$ and 2, as we seek to take the first step to evaluate the classification ability of MICS. This also reflects the most common cases that our subject MICSs should properly handle, as in *VOC* and *COCO*, about 42% of images are annotated with one label and 73% of images are annotated with no more than two labels. Moreover, we note that modern DNN-based MICSs are usually developed by the deep embedding learning method [61], in which the relationships between each pair of labels in the training set are exploited to perform prediction. As such, the testing of $k = 1$ and 2 also indicates the evaluation of the most basic functionality of MICSs in handling correlations between labels.

1) Process for RQ_1: ATOM relies on image search engine and natural language processing to collect at most five test images for each common k-label combination l. However, due to the search accuracy, the k objects specified in l might not always be present in those test images collected. Hence, in the first research question, we will investigate to what extent the test image generation process of ATOM can produce test images that are *relevant* to common k-label combinations.

Specifically, we first applied ATOM to automatically collect test images for the three label spaces in Table II. This results in a total number of 17,160 test images. Then, we managed to manually annotate these images to determine whether each of them is indeed *relevant* to the corresponding common k-label combination. To this end, we set a guideline that classifies each test image into one of the three groups: 1) *Strong-Relevant*, for which the image clearly contains all the k objects in l; 2) *Weak-Relevant*, for which the image contains all the k objects, but with a certain degree of vagueness; 3) *Irrelevant*, for which at least one of the k objects is clearly missing (i.e., invalid for testing l). From a practical viewpoint, as long as a human can confidently recognize a specific object in an image, then a powerful MICS should also be able to recognize it, even if this object is a little bit vague in its shape. Hence, test images in both *Strong-Relevant* and *Weak-Relevant* groups are considered as *relevant* test images, i.e., valid test inputs for testing MICS [62].

To ease the understanding, we have identified four cases that might make an image in the *Weak-Relevant* group, as illustrated in Fig. 3: 1) *Blur*, the object is too small, or with low resolution; 2) *Incompleteness*, some features of the object are not complete; 3) *Merging*, multiple objects are combined into one, e.g., an airplane-shaped cake; and 4) *Abstraction*, relevant but not real-world object is contained, e.g., a cartoon person. Our annotation guideline indicates that all of the k objects should still be confidently recognized by humans for those weak-relevant images. We note that such images are also common in the *VOC* and *COCO* datasets[3], so they do indicate valid test inputs for testing MICS.



(a) Blur ($\{airplane, bird\}$) (b) Incompleteness ($\{bottle, table\}$)

(c) Merging ($\{airplane, cake\}$) (d) Abstraction ($\{cow, person\}$)

Fig. 3. The four cases that might make a test image in the *Weak-Relevant* group (the corresponding label combination is given in parenthesis).

In this study, to avoid potential subjective bias of annotation, each test image is independently annotated by three students majoring in computer science. To increase the agreement of the decision, we have first scanned all images in the original testing sets of the three datasets to better understand the semantic information of each label. Next, these students are assigned a set of 500 randomly sampled test images for annotation, and a meeting is then held to calcify misunderstandings and reach a consensus (especially for the images in the *Weak-Relevant* group). After this step, these students independently annotated the remaining images, and the final result is obtained via voting (if the three results differ from each other, then the image is annotated as *Weak-Relevant*). Finally, we have achieved a *Cohen's Kappa* agreement rate of 0.953 (an almost perfect level of agreement [63]) across all test images collected.

2) Process for RQ_2: The second research question concerns the effectiveness of ATOM in revealing potential errors in MICS. According to the oracle identification process of ATOM (Section II-C), each test image will be classified into the three groups: *confident images* (C), *label error images* (E), and *unspecific error images* (U). In this experiment, we directly used all test images collected in RQ_1 as the test inputs to obtain the three groups for each subject MICS.

Here, test images in both E and U are considered as *error-revealing images*, as they both fail metamorphic testing (i.e., violate at least one MR). We note that this will not lead to any false positive, because as long as the MICS cannot produce identical outputs for both source and follow-up test images, then by the definition of MR, there must be an error in MICS (no matter whether the objects are correctly recognized).

Despite that both E and U indicate potential errors, in this study, we focus more on images in E, because such errors

[3]For example, *Blur* (http://cocodataset.org/#explore?id=160626), *Incompleteness* (id=273416), *Merging* (id=293011), and *Abstraction* (id=9479) images in the training/validation set of *COCO*.

Table III: The Number of All ($|L_k|$) and Common ($|L|$) k-Label Combinations, Number of Test Images Collected and Test Images of Different Groups, and Proportion of Relevant Test Images (*RI*) and Relevant Label Combinations (*RLC*)

k	Dataset	$\|L_k\|$	$\|L\|$	# Test Images	# *Strong-Relevant*	# *Weak-Relevant*	# *Irrelevant*	*RI*	*RLC*
1	VOC	20	20	100	100	0	0	100%	100%
	COCO	80	80	400	390	5	5	98.8%	98.8%
	PHOTO	32	32	160	159	1	0	100%	100%
2	VOC	190	188	898	600	180	118	86.9%	98.4%
	COCO	3,160	2,951	13,297	5,519	4,600	3,178	76.1%	89.8%
	PHOTO	496	483	2,305	1,172	809	324	85.9%	97.7%

are considered relevant to the classification ability of label combinations. However, due to the search accuracy, irrelevant images might be used, and thus lead to potential false positives in the results reported by ATOM. For example, ATOM might collect a test image that contains a *cat* only for $l = \{cat, dog\}$. Assuming that the output on the source test image is $\{cat\}$, and the output on all follow-up test images is $\{dog\}$. In this case, ATOM will classify this test image as a *label error image* (i.e., the error is due to the classification of l), but this is a false positive, because this test image is an invalid test input for testing l. To account for this, we further relied on the annotation results obtained in RQ_1 to calculate the errors introduced by such invalid test images in the final results of ATOM (in particular, the E set).

In addition, we further compared the error-revealing ability of ATOM with a random testing approach (as to the best of our knowledge, there is no black-box testing approach for MICS). Specifically, the random approach is unaware of the label space, and selects test images randomly from the testing set of *ImageNet* [53]. This testing set contains 100,000 images, and is not used for the pre-training of any subject DNN model included in this study. We let the random approach select and test the same number of test images as that of ATOM. Since no label information is provided for the random approach, its test oracle is identified by metamorphic testing only (using the same MRs of ATOM, i.e., those in Table I). The execution of the random approach is repeated five times for each DNN to account for the randomness involved.

3) Process for RQ_3: Finally, in the last research question, we based on the testing results of ATOM to investigate the ability of current MICS in classifying different label combinations. Specifically, for each common k-label combination l, ATOM will produce a set of confident images, C_l, and a set of label error images, E_l (note that C_l and E_l contain source test images only). These two sets exactly indicate the two cases where the MICS has a good and poor classification ability on l, respectively. Then, we based on these two sets and the classification outputs of both source and follow-up images to calculate the *classification score* of l to measure the ability of MICS in classifying l:

$$\text{score}(l) = \frac{|\{x_i \mid l \subseteq f(x_i), x_i \in E_l + C_l, 0 \leq i \leq 7\}|}{|E_l + C_l| * 8} \quad (1)$$

The above score calculates the proportion of test images in which all k objects involved in l can be correctly recognized, i.e., $l \subseteq f(x_i)$, among all source and follow-up images used for testing l. Note that ATOM uses seven MRs, and there will be one source image x_0, and seven follow-up images, $\{x_1, \ldots, x_7\}$. So there will be a total number of $|E_l + C_l| * 8$ images here. Clearly, a larger $\text{score}(l)$ indicates a better classification ability on the label combination l.

For example, assume that ATOM produces $C_l = \{x^1\}$ and $E_l = \{x^2\}$ for $l = \{cat, dog\}$. By the definition of C_l, we must have $l \subseteq f(x_i^1)$ for $0 \leq i \leq 7$. Assuming that for x^2, $f(x_0^2) = f(x_1^2) = l$ and $f(x_i^2) = \{cat\}$ for $2 \leq i \leq 7$ (i.e., among the eight images, only two are correctly classified). We thus have $\text{score}(l) = (8 + 2)/(2 \times 8) = 0.625$.

The above experiments were performed on a machine with Intel(R) Xeon(R) CPU E5-2620, GeForce RTX 2080Ti GPU, and 32GB RAM. We note that the testing process of ATOM is fully automated, and it takes about 10 seconds to test each label combination in our experiments.

IV. Results

A. Results for RQ_1 (relevant test images)

The first research question investigates the effectiveness of ATOM in collecting test images that are relevant to common k-label combinations. Table III gives the number of all and common k-label combinations of each dataset, as well as the number of test images collected and the accuracy achieved. Specifically, we report the number of images that are manually classified into the three groups (*Strong-Relevant*, *Weak-Relevant*, and *Irrelevant*). We then based on these values to calculate the proportion of *relevant test images* (i.e., images in *Strong-Relevant* and *Weak-Relevant*), *RI*, and the proportion of common k-label combinations in which at least one test image is identified as relevant (*relevant label combinations*, *RLC*).

From Table III, we can see that ATOM collects 660 and 16,500 test images in total for $k = 1$ and 2, respectively. For $k = 1$, ATOM is effective in collecting relevant test images, as both *RI* and *RLC* can achieve 100% for *VOC* and *PHOTO* (especially, more than 99.4% of these images are classified as *Strong-Relevant*). This indicates that all test images collected for these two datasets are indeed relevant to their corresponding label combinations. For *COCO*, five test images (1.3%) collected by ATOM are considered irrelevant. This is because the *keyboard* label's super category is *electronic* in this dataset, and the inclusion of this super category tends to mislead the search (an electronic piano, rather than a keyboard of a computer, is usually found).

Table IV: The Number of Label Error Images ($|E|$) and Unspecific Error Images ($|U|$), and Proportion of Confident Label Combinations ($|L_c|/|L|$) and Vulnerable Label Combinations ($|L_e|/|L|$) (Values in Parentheses Indicate the Errors Introduced by Irrelevant Images Collected)

k	Dataset	MICS	$\|E\|$	$\|U\|$	$\|L_c\|/\|L\|$	$\|L_e\|/\|L\|$
1	VOC	MSRN	3 (0)	10	100% (0%)	15% (0%)
		MLGCN	5 (0)	12	100% (0%)	20% (0%)
		DSDL	8 (0)	9	100% (0%)	30% (0%)
		ASL	4 (0)	5	100% (0%)	15% (0%)
		MCAR	6 (0)	20	100% (0%)	25% (0%)
	COCO	MSRN	21 (0)	70	97.5% (0%)	22.5% (0%)
		MLGCN	18 (0)	95	96.3% (0%)	18.8% (0%)
		DSDL	29 (0)	140	92.5% (0%)	26.3% (0%)
		ASL	7 (0)	52	98.8% (1.3%)	6.3% (0%)
		MLD	3 (1)	112	98.8% (1.3%)	3.8% (1.3%)
	PHOTO	-	13 (0)	5	93.8% (0%)	28.1% (0%)
2	VOC	MSRN	113 (3)	191	44.1% (1.1%)	42% (1.1%)
		MLGCN	103 (4)	199	32.4% (0%)	41% (1.6%)
		DSDL	120 (5)	221	33.5% (0.5%)	42.6% (1.6%)
		ASL	57 (4)	136	49.5% (0%)	23.4% (1.6%)
		MCAR	168 (8)	284	41% (0%)	55.3% (2.1%)
	COCO	MSRN	1,126 (16)	4,858	28.5% (0.1%)	28.1% (0.4%)
		MLGCN	1,217 (30)	5,267	27.9% (0.1%)	30.1% (0.6%)
		DSDL	1,444 (40)	6,725	27.2% (0.2%)	34.6% (1.1%)
		ASL	807 (11)	3,777	38% (0.2%)	22.1% (0.2%)
		MLD	861 (57)	7,165	49.3% (0.8%)	23.7% (1.5%)
	PHOTO	-	33 (1)	536	2.3% (0.2%)	5.8% (0.2%)

Table V: The Proportion of Distinct k-Label Combinations Included in the Outputs of All Test Images and Error-Revealing Images of the ATOM and Random Approaches

k	Dataset	MICS	All Test Images		Error-Revealing Images	
			ATOM	Random	ATOM	Random
1	VOC	MSRN	100%	83%	55%	71%
		MLGCN	100%	89%	60%	78%
		DSDL	100%	83%	75%	71%
		ASL	100%	80%	55%	60%
		MCAR	100%	87%	70%	79%
	COCO	MSRN	100%	91.3%	78.8%	86.3%
		MLGCN	100%	93.5%	86.3%	89.5%
		DSDL	100%	96.3%	97.5%	95.5%
		ASL	100%	92%	67.5%	84.5%
		MLD	100%	99.5%	85%	99%
	PHOTO	-	100%	49.4%	37.5%	40%
2	VOC	MSRN	64.7%	28.3%	56.3%	26.6%
		MLGCN	60%	31.3%	54.7%	29.5%
		DSDL	58.9%	24.5%	52.6%	23.4%
		ASL	62.1%	23.5%	40%	20.3%
		MCAR	80%	42.2%	77.4%	41.3%
	COCO	MSRN	57.6%	36.7%	53.6%	36.2%
		MLGCN	64.3%	42.8%	60.3%	42.3%
		DSDL	84.5%	64.3%	82.6%	64.1%
		ASL	64.8%	40%	57.3%	38.8%
		MLD	96.5%	87.6%	95.9%	87.5%
	PHOTO	-	9.9%	6.3%	9.7%	6.3%

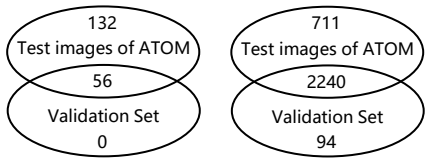
In contrast to 1-label combinations, collecting test images that contain two specific objects is more challenging, as both *RI* and *RLC* tend to decrease for $k = 2$ (meanwhile, more test images tend to be classified as *Weak-Relevant*). Nevertheless, ATOM can still achieve a maximum *RI* of 86.9% (for *VOC*), and the average *RI* observed across all three datasets is 83%. Moreover, the *RLC* achieved is relatively higher, and ATOM can collect at least one relevant test image for at least 89% of common 2-label combinations.

Answer to $\mathbf{RQ_1}$: ATOM is effective in collecting test images that are relevant to each label combination. On average, ATOM can collect at least one relevant test image for 99.6% and 95.3% of common 1- and 2-label combinations, respectively.

B. Results for RQ_2 (error revealing)

The second research question investigates the error revealing ability of ATOM. Table IV gives the number of error-revealing images (including *label error images*, $|E|$, and *unspecific error images*, $|U|$), and proportion of *confident label combinations*, $|L_c|/|L|$, and *vulnerable label combinations*, $|L_e|/|L|$ obtained on each MICS (according to the results of test oracle identification, Algorithm 1).

From Table IV, we can see that ATOM yields a total number of 647 and 35,408 error-revealing images for 1- and 2-label combinations (20.9% and 41.1% of all test images collected), respectively. Among these test images, 117 and 6,049 images are identified as *label error images*, E (i.e., the error is related to the classification of the label combination, and is thus the focus of ATOM). In general, the current MICSs perform well when handling each single object ($k = 1$). They can correctly classify at least one test image for, on average, 98% of common 1-label combinations (i.e., confident label combinations, L_c), and there are only about 19% of common 1-label combinations in which *label error images* are identified (i.e., vulnerable label combinations, L_e). However, when testing 2-label combinations, we can see that the performance of MICSs dramatically decreases. Especially, the average proportion of confident label combinations is only 34%, and the proportion of vulnerable label combinations can be up to 55.3%. This finding reveals that the current MICSs remain ineffective in handling correlations between even two objects.

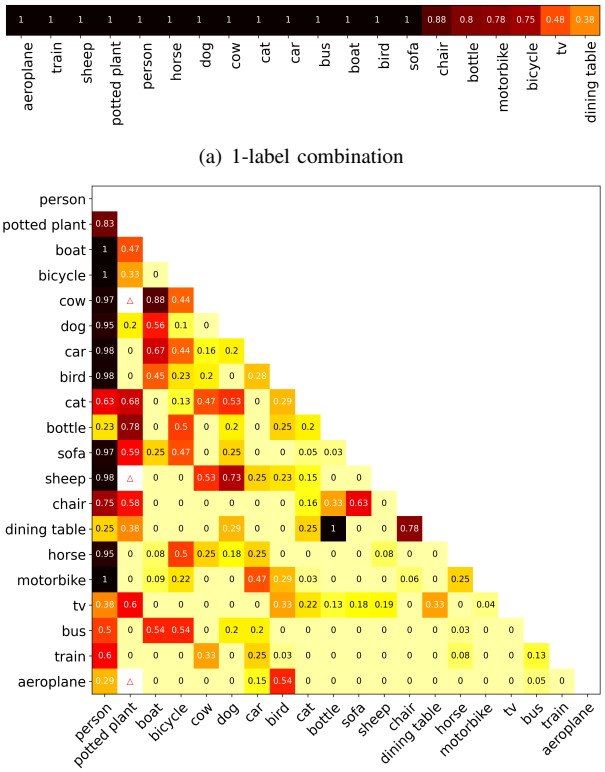


(a) *VOC* (b) *COCO*

Fig. 4. The relationship of 2-label combinations that appear in the validation sets of open-source datasets and relevant test images collected by ATOM.

Note that due to the irrelevant test images collected in ATOM, there might be false positives in the *label error images* reported (as discussed in Section III-B2). To this end, Table IV further gives the errors introduced in E, $|L_c|/|L|$, and $|L_e|/|L|$ by such irrelevant test images, as shown in the corresponding parentheses of each data cell. From these data, we can see that the errors introduced in ATOM are generally small. For $k = 1$, there is at most one test image is misclassified in E (i.e., for *MLD* on *COCO*), and the error in $|L_c|/|L|$ and $|L_e|/|L|$ is at most 1.3%. Even for $k = 2$, the maximum errors introduced to E, $|L_c|/|L|$, and $|L_e|/|L|$ are only 57 (for *MLD* on *COCO*), 1.1% (for *MSRN* on *VOC*), and 2.1% (for *MCAR* on *VOC*), respectively.

In particular, for the industrial MICS product, *PHOTO*, ATOM reports 587 error-revealing images. By the definition



(a) 1-label combination

(b) 2-label combination

Fig. 5. The classification score of *MLGCN* for the *VOC* dataset.

of MR, all of these images indicate potential errors of MICS (as different outputs are observed on their follow-up images). We have sent all of these images to our industrial partner. Then, a senior engineer of the testing team took several days to manually check these images, and confirmed 92% of them. The main reason that some error-revealing images are not acknowledged is due to the specific requirements of *PHOTO*, that is, there are some scenarios that a partial correct classification is acceptable. For example, in some cases, as long as the most salient object (the object is in the center of the image and large in size) can be recognized, then the classification is considered correct (even if some other relatively small objects can also be easily recognized by humans).

Moreover, note that all of the subject MICSs can achieve at least 81% mean average precision in their respective validation sets (see Table II), but ATOM can still find tens of thousands of error-revealing images. To investigate the potential reasons, we further analyzed the k-label combinations that appear (i.e., are covered) in the validation set of the two open-source datasets, and the set of *relevant* test images collected by ATOM. We find that all common 1-label combinations are covered by both of these two sets. While for $k = 2$, the relationship of these two sets is shown in Fig. 4. From Fig. 4, we can see that the relevant test images collected by ATOM cover a substantial number of 2-label combinations that do not appear in the validation sets of both datasets. For *VOC*, there are 132 (70% of all common 2-label combinations) potential correlations between two objects are actually tested by ATOM but not the validation set, and this number is 711 (24%) for *COCO*. This finding reveals that the current validation sets might not reflect the data distribution in the real-world.

At last, we compare ATOM with the random approach. Table V gives the proportion of distinct k-label combinations included in the outputs of all test images and error-revealing images of the ATOM and random approaches. Since different states or paths of the DNN are typically triggered when the MICS produces different outputs, from Table V, we can see that the test images collected by ATOM tend to trigger more diverse behaviors of the MICS than the randomly collected images. However, for the 1-label combinations included in error-revealing images, the average proportion achieved by the random approach is 7.8% higher than that of ATOM. This is because test images collected by ATOM for $k = 1$ tend to contain one object only, and those images are relatively easy to classify (unlikely to reveal errors). While for $k = 2$, ATOM tends to examine more diverse error scenarios, as 20.4% more 2-label combinations are included in error-revealing images of ATOM than in those of the random approach.

Answer to $\mathbf{RQ_2}$: ATOM reveals a total number of 117 and 6,049 label error images when examining 1- and 2-label combinations of the MICSs studied, respectively. The label combination indicates an effective test criterion for MICS, as there are 70% and 24% of common 2-label combinations that appear in the test images of ATOM but not in the validation sets of the two open-source datasets, *VOC* and *COCO*.

C. Results for RQ_3 (classification ability)

The last research question investigates the concrete classification ability of MICS on different common k-label combinations. Figs. 5(a) and 5(b) show the classification score of *MLGCN* for *VOC* (computed based on Equation 1) for $k = 1$ and 2, respectively. Here, the triangle in the figure indicates that there is no confident image and label error image for the label combination l (i.e., $|E_l + C_l| = 0$), and so no score can be obtained. Due to the space limitation, the results of the other nine MICSs (except for *PHOTO*) are provided at: https://github.com/GIST-NJU/ATOM.

From Fig. 5(a), we can see that *MLGCN* achieves a classification score of 1.0 in 70% of cases, indicating that this MICS has a reasonably good classification ability in recognizing each single object of its label space. However, from Fig. 5(b), the classification score is zero in 48% of cases (i.e., no image can be correctly classified). As a similar result is also observed for other MICSs studied, current MICSs remain ineffective in handling correlations between even two objects.

Despite that the classification ability of *MLGCN* is generally weak for $k = 2$, from Fig. 5(b), we can still find some 2-label combinations that *MLGCN* can properly handle (e.g., the label combinations that contain a *person*). We have further analyzed the images in the training set of *VOC*, and we find

(a) $l = \{bicycle, bottle\}$
$f(x) = \{bicycle, person\}$

(b) $l = \{bus, skateboard\}$
$f(x) = \{bus, person\}$

(c) $l = \{\text{bicycle}, sofa\}$
$f(x) = \{\text{bicycle}, person\}$

(d) $l = \{\text{tv}, car\}$
$f(x) = \{\text{tv}, chair\}$

(e) $l = \{\text{chair}, toothbruth\}$
$f(x) = \{\text{chair}, dining\ table\}$

(f) $l = \{\text{apple}, book\}$
$f(x) = \{\text{apple}, knife\}$

Fig. 6. The three typical cases (the three rows) that might result in label combination related errors of MICS.

that there are typically more test images for those 2-label combinations. For example, for the 2-label combinations that achieve classification scores of 1.0 and 0.98 (a total number of four and three label combinations), there is an average number of 104.8 and 83 relevant test images in the training set, respectively. While for the 2-label combinations with scores 0 and 0.03 (a total number of 89 and four label combinations), the average number of relevant test images is only 1.9 and 2, respectively. This finding reveals that the classification score reported by ATOM could imply the data distribution in the training set, and could thus provide hints for developers to improve the data quality of MICSs.

Moreover, we randomly sampled 1,000 distinct test images from all *label error images* produced by ATOM, and manually analyzed these images to summarise three typical cases that label combination related errors might occur (i.e., the MICS is unable to correctly recognize all of the k objects involved in the label combination). Fig. 6 uses six test images to illustrate these cases, where l and $f(x)$ indicate the label combination used for the image search and the classification output, respectively. For the first case, the error is likely to be triggered when there is a large difference in the sizes of different objects (e.g., the *bottle* and *bicycle* in Fig. 6(a)). For the second case, the error is likely to be triggered when some of the objects are displayed in a different form than the usual scene (e.g., in Fig. 6(c), the back of the *sofa* is displayed with a *bicycle* in the image, which differs from the case that the front or side of the *sofa* is usually displayed). For the third case, the error is likely due to the incorrect correlations captured by the MICS (e.g., there is only a *chair* in Fig. 6(e), but a *dining table* is predicted).

We note that ATOM aims to test each k-label combination l independently, so that whether there is a correct classification is determined based on the recognition of l (i.e., all labels in l should be recognized). So, for the first image that contains the three objects, $\{bicycle, bottle, person\}$, in Fig. 6, the classification output $f(x) = \{bicycle, person\}$ is considered incorrect, because the current focus is on the $bottle$ instead of the $person$ (as such, the classification ability on $\{bicycle, bottle\}$ can be evaluated based on this image). We note that this image is also valid for testing the 2-label combination $\{bicycle, person\}$, and at this time, the above classification output is considered correct.

> **Answer to RQ$_3$:** The current MICS remains ineffective even in classifying 2-label combinations, as a zero classification score is observed in about half of cases.

V. Threats to Validity

Regarding the internal threats to validity, the experimental results reported in this study might be influenced by the metamorphic relations (MRs) and their parameter values used. We note that all the seven MRs used are common image processing operations [44], [45], and we have used parameter values that introduce as fewer perturbations as possible to preserve the semantics of test images. Meanwhile, the results reported might also be influenced by the manual annotation of the test images collected. To mitigate this risk, we have set guidelines to specify the criteria and examples, and each test image was independently annotated by three students (with an agreement rate of 0.953). We have made all the test images collected (with annotations), and the code of generating follow-up images publicly available (for the two open-source datasets), so that others can check and reuse them.

In addition, in this study, we have applied ATOM to test k-label combinations for $k = 1$ and 2 only, and there might be a limitation on the value of k that ATOM can efficiently handle. We acknowledge that the number of k-label combinations will grow exponentially with the increase of k, so ATOM tends to be hard to scale when k keeps growing (especially for $k > 4$). But, at the same time, the number of rare k-label combinations (i.e., the label combinations that are not common to the real-world) will also greatly increase with the increase of k. Accordingly, iterating and testing every k-label combination becomes inefficient and unnecessary. In this case, ATOM can be used in an incremental manner, in which only those k-label combinations that the MICS can correctly classify will be extended to the corresponding $(k+1)$-label combinations. According to our experimental results, the current MICS remains ineffective even in classifying 2-label combinations. This also suggests that the quality of current MICSs should be improved

first before moving to the examination of larger values of k. Nevertheless, ATOM is a fully automated approach, and testers can specify any k-label combination in which they are interested as the test target of ATOM.

As far as external threats to validity are concerned, we have evaluated ATOM on 11 MICSs only. ATOM might thus exhibit different performance when testing different MICSs. Nevertheless, both *VOC* [39] and *COCO* [40] indicate popular and widely used datasets of MICS, and we have used five state-of-the-art DNNs for each of them. We have also included a real-world industrial product, *PHOTO*, in the experiments. We believe that these MICSs are representative subjects to evaluate the effectiveness of ATOM.

VI. Related Work

To develop more appropriate test coverage criteria for DNN-based systems, Pei et al. [12] proposed neuron coverage to measure the proportion of neurons that can be activated by the given test inputs. This criterion was then extended by considering different network structures and granularity of coverages [13]–[18], but meanwhile, the relations of such criteria and test effectiveness were questioned by some other studies [64]–[67]. In addition to the network structure, Kim et al. [19] exploited the training data of DNNs to propose surprise adequacy, which measures the similarity of system behaviors between test inputs and training data. There are also studies [68] that propose to use mutation score as the coverage criterion. In addition to the above white-box based criteria, there are also black-box based criteria that rely on neither network structure nor training data of DNNs. For example, Byun et al. [20], [69] proposed manifold combination coverage, which is based on the semantics of the input space compressed by manifold learning. Aghababaeyan et al. [21] relied on a pre-trained feature extraction model to design three diversity metrics for testing image recognition systems.

For the test input generation of DNN-based systems, a common strategy is to generate test inputs that can maximize some pre-defined coverage criteria [12], [24]–[28], [70], [71]. To this end, techniques like search algorithm [12], mutation and transformation based [28], fuzzing [24]–[27], and symbolic execution [70], [71] have been extensively explored. By contrast, some other studies [37], [72]–[75] seek to utilize adversarial input generation to find error-revealing test inputs. For example, Zhang et al. [37] and Zhou et al. [72] proposed to generate synthesized driving scenes to test autonomous driving systems. Narodytska and Kasiviswanathan [73] relied on the probability distribution of the output to generate adversarial inputs for convolutional neural networks.

Finally, for the test oracle identification, metamorphic testing is one of the most widely used choices for testing DNN-based systems [26], [28], [33]–[38], in which various metamorphic relations have been proposed for different application domains. Another common choice is to use differential testing [12], [17], [29]–[32], and it requires the availability of multiple system implementations.

The ATOM framework presented in this study is specially designed for testing DNN-based MICSs. It differs from the previous studies in three main aspects. First, ATOM uses label combination as the coverage criterion, which does not rely on the internals of DNN, and takes the core functionality of MICS into account. This differs from white-box based criteria [12]–[19], [68], as well as general black-box based criteria for DNNs [20], [21], [69]. Second, ATOM adopts image search engine to collect realistic test inputs, which differs from those that rely on seed based [12], [24]–[28], [70], [71] or adversarial input based generation [37], [72]–[75]. Despite that the image search engine has also been used by Wan et al. for testing image classification API [76], ATOM further manages to filter out irrelevant images by natural language processing. Finally, ATOM combines metamorphic testing and label information to identify test oracle, and is, to the best of our knowledge, the first approach to evaluate the classification ability of MICS on different label combinations.

VII. Conclusion

In contrast to the testing of traditional software systems, the different and unique nature of DNN-based systems especially ask for different testing strategies. In this paper, we propose ATOM, an automated black-box testing framework for MICS, which takes the concept of k-label combination as the coverage criterion to systematically examine the correlations of k objects in the classification accuracy. ATOM then relies on image search engine and natural language processing to find realistic and relevant test images as the test inputs, and finally combines metamorphic testing and label information to realize test oracle identification. Experimental results reveal that ATOM is effective in revealing potential errors in current DNN-based MICSs, and the classification ability of those MICSs remains ineffective even in classifying 2-label combinations only.

In order to aid others to replicate and extend the experimental results of this study, we provide the source code of ATOM, all test images collected (with annotations), and the code of generating follow-up images for the two open-source datasets (*VOC* and *COCO*) at: https://github.com/GIST-NJU/ATOM.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62072226 and 62102176), and the Natural Science Foundation of Jiangsu Province (No. BK20221439).

References

[1] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *Trans. Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Anal.*, vol. 42, no. 12, pp. 60–88, 2017.

[3] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2016, pp. 817–825.

[4] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285–2294.

[5] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proceedings of the International Conference on Computer Vision*, 2017, pp. 464–472.

[6] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2027–2036.

[7] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[8] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16 478–16 488.

[9] Z.-M. Chen, Q. Cui, B. Zhao, R. Song, X. Zhang, and O. Yoshie, "Sst: Spatial and semantic transformers for multi-label image recognition," *Trans. Image Process.*, vol. 31, pp. 2570–2583, 2022.

[10] L. Grush, "Google engineer apologizes after photos app tags two black people as gorillas," 2017, https://www.theverge.com/2015/7/1/8880363/googleapologizes-photos-app-tags-two-black-people-gorillas.

[11] N. E. Boudette., "Tesla's self-driving system cleared in deadly crash," 2015, https://nyti.ms/2iZ93SL.

[12] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated whitebox testing of deep learning systems," in *Proceedings of the Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[13] L. Ma, F. Juefei-Xu, F. Zhang, J. Sun, M. Xue, B. Li, C. Chen, T. Su, L. Li, Y. Liu, J. Zhao, and Y. Wang, "Deepgauge: multi-granularity testing criteria for deep learning systems," in *Proceedings of the International Conference on Automated Software Engineering*, 2018, pp. 120–131.

[14] Y. Sun, X. Huang, and D. Kroening, "Testing deep neural networks," *CoRR*, vol. abs/1803.04792, 2018.

[15] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, "Structural test coverage criteria for deep neural networks," *Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, 2019.

[16] L. Ma, F. Juefei-Xu, M. Xue, B. Li, L. Li, Y. Liu, and J. Zhao, "Deepct: Tomographic combinatorial testing for deep learning systems," in *Proceedings of the International Conference on Software Analysis, Evolution and Reengineering*, 2019, pp. 614–618.

[17] J. Sekhon and C. Fleming, "Towards improved testing for deep learning," in *Proceedings of the International Conference on Software Engineering: New Ideas and Emerging Results*, 2019, pp. 85–88.

[18] S. Gerasimou, H. F. Eniser, A. Sen, and A. Çakan, "Importance-driven deep learning system testing," in *Proceedings of the International Conference on Software Engineering*, 2020, pp. 702–713.

[19] J. Kim, R. Feldt, and S. Yoo, "Guiding deep learning system testing using surprise adequacy," in *Proceedings of the International Conference on Software Engineering*, 2019, pp. 1039–1049.

[20] T. Byun, S. Rayadurgam, and M. P. E. Heimdahl, "Black-box testing of deep neural networks," in *Proceedings of the International Symposium on Software Reliability Engineering*, 2021, pp. 309–320.

[21] Z. Aghababaeyan, M. Abdellatif, L. Briand, S. Ramesh, and M. Bagherzadeh, "Black-box testing of deep neural networks through test case diversity," *Trans. Softw. Eng.*, vol. 49, no. 05, pp. 3182–3204, 2023.

[22] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 999–1008.

[23] Y. Tian, S. Ma, M. Wen, Y. Liu, S. Cheung, and X. Zhang, "To what extent do dnn-based image classification models make unreliable inferences?" *Empir. Softw. Eng.*, vol. 26, no. 4, p. 84, 2021.

[24] J. Guo, Y. Jiang, Y. Zhao, Q. Chen, and J. Sun, "Dlfuzz: Differential fuzzing testing of deep learning systems," in *Proceedings of the Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 739–743.

[25] A. Odena, C. Olsson, D. Andersen, and I. Goodfellow, "Tensorfuzz: Debugging neural networks with coverage-guided fuzzing," in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 4901–4911.

[26] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See, "Deephunter: a coverage-guided fuzz testing framework for deep neural networks," in *Proceedings of the International Symposium on Software Testing and Analysis*, 2019, pp. 146–157.

[27] S. Demir, H. F. Eniser, and A. Sen, "Deepsmartfuzzer: Reward guided test generation for deep learning," in *Proceedings of the Workshop on Artificial Intelligence Safety*, 2020, pp. 134–140.

[28] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the International Conference on Software Engineering*, 2018, pp. 303–314.

[29] C. Murphy, G. E. Kaiser, and M. Arias, "An approach to software testing of machine learning applications," in *Proceedings of the International Conference on Software Engineering & Knowledge Engineering*, 2007, p. 167.

[30] C. Murphy, G. Kaiser, and M. Arias, "Parameterizing random test data according to equivalence classes," in *Proceedings of the international workshop on Random testing*, 2007, pp. 38–41.

[31] Y. Qin, H. Wang, C. Xu, X. Ma, and J. Lu, "Syneva: Evaluating ml programs by mirror program synthesis," in *Proceedings of the International Conference on Software Quality, Reliability and Security*, 2018, pp. 171–182.

[32] X. Xie, L. Ma, H. Wang, Y. Li, Y. Liu, and X. Li, "Diffchaser: Detecting disagreements for deep neural networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019, pp. 5772–5778.

[33] C. Murphy, G. E. Kaiser, L. Hu, and L. Wu, "Properties of machine learning applications for use in metamorphic testing," in *Proceedings of the Twentieth International Conference on Software Engineering & Knowledge Engineering*, 2008, pp. 867–872.

[34] C. Murphy, K. Shen, and G. Kaiser, "Using jml runtime assertion checking to automate metamorphic testing in applications without test oracles," in *Proceedings of the International Conference on Software Testing Verification and Validation*, 2009, pp. 436–445.

[35] Murphy, Christian and Shen, Kuang and Kaiser, Gail, "Automatic system testing of programs without test oracles," in *Proceedings of the International Symposium on Software Testing and Analysis*, 2009, pp. 189–200.

[36] J. Ding, X. Kang, and X.-H. Hu, "Validating a deep learning framework by metamorphic testing," in *Proceedings of the International Workshop on Metamorphic Testing*, 2017, pp. 28–34.

[37] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, "Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems," in *Proceedings of the International Conference on Automated Software Engineering*, 2018, pp. 132–142.

[38] A. Dwarakanath, M. Ahuja, S. Sikand, R. M. Rao, R. P. J. C. Bose, N. Dubash, and S. Podder, "Identifying implementation bugs in machine learning based image classifiers using metamorphic testing," in *Proceedings of the International Symposium on Software Testing and Analysis*, 2018, pp. 118–128.

[39] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[40] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[41] C. Nie and H. Leung, "A survey of combinatorial testing," *Comput. Surv.*, vol. 43, no. 2, pp. 11:1–11:29, 2011.

[42] E. Lanus, L. J. Freeman, D. R. Kuhn, and R. N. Kacker, "Combinatorial testing metrics for machine learning," in *Proceedings of the International Conference on Software Testing, Verification and Validation Workshops*, 2021, pp. 81–84.

[43] Z. Zhang, P. Wang, H. Guo, Z. Wang, Y. Zhou, and Z. Huang, "Deepbackground: Metamorphic testing for deep-learning-driven image recognition systems accompanied by background-relevance," *Inf. Softw. Technol.*, vol. 140, p. 106701, 2021.

[44] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, p. 60, 2019.

[45] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," in *Proceedings of the International Conference on Learning Representations*, 2017.

[46] K. Pei, Y. Cao, J. Yang, and S. Jana, "Towards practical verification of machine learning: The case of computer vision systems," *CoRR*, vol. abs/1712.01785, 2017.

[47] X. Qu, H. Che, J. Huang, L. Xu, and X. Zheng, "Multi-layered semantic representation network for multi-label image classification," *Int. J. Mach. Learn. Cybern.*, pp. 1–9, 2023.

[48] Z. Chen, X. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[49] F. Zhou, S. Huang, and Y. Xing, "Deep semantic dictionary learning for multi-label image classification," in *Proceedings of the Conference on Artificial Intelligence, Conference on Innovative Applications of Artificial Intelligence, Symposium on Educational Advances in Artificial Intelligence*, 2021, pp. 3572–3580.

[50] T. Ridnik, E. B. Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," in *Proceedings of the International Conference on Computer Vision*, 2021, pp. 82–91.

[51] B. Gao and H. Zhou, "Learning to discover multi-class attentional regions for multi-label image recognition," *Trans. Image Process.*, vol. 30, no. 6, pp. 5920–5932, 2021.

[52] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, and A. Noy, "Ml-decoder: Scalable and versatile classification head," in *Proceedings of the Winter Conference on Applications of Computer Vision*, 2023, pp. 32–41.

[53] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[54] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *Int. J. Comput. Vis.*, 2020.

[55] ASL, https://github.com/Alibaba-MIIL/ASL.

[56] DSDL, https://github.com/ZFT-CQU/DSDL.

[57] MCAR, https://github.com/gaobb/MCAR.

[58] MLGCN, https://github.com/Megvii-Nanjing/ML-GCN.

[59] MSRN, https://github.com/chehao2628/MSRN.

[60] MLD, https://github.com/alibaba-miil/ml_decoder.

[61] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The emerging trends of multi-label learning," *Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7955–7974, 2022.

[62] V. Riccio and P. Tonella, "When and why test generators for deep learning produce invalid inputs: an empirical study," in *Proceedings of the International Conference on Software Engineering*, 2023, pp. 1161–1173.

[63] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia medica*, vol. 22, no. 3, pp. 276–282, 2012.

[64] Z. Li, X. Ma, C. Xu, and C. Cao, "Structural coverage criteria for neural networks could be misleading," in *Proceedings of the International Conference on Software Engineering: New Ideas and Emerging Results*, 2019, pp. 89–92.

[65] Y. Dong, P. Zhang, J. Wang, S. Liu, J. Sun, J. Hao, X. Wang, L. Wang, J. S. Dong, and T. Dai, "An empirical study on correlation between coverage and robustness for deep neural networks," in *Proceedings of the International Conference on Engineering of Complex Computer Systems*, 2020, pp. 73–82.

[66] J. Chen, M. Yan, Z. Wang, Y. Kang, and Z. Wu, "Deep neural network test coverage: How far are we?" *CoRR*, vol. abs/2010.04946, 2020.

[67] F. Harel-Canada, L. Wang, M. A. Gulzar, Q. Gu, and M. Kim, "Is neuron coverage a meaningful measure for testing deep neural networks?" in *Proceedings of the Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2020, pp. 851–862.

[68] W. Shen, J. Wan, and Z. Chen, "Munn: Mutation analysis of neural networks," in *Proceedings of the International Conference on Software Quality, Reliability and Security Companion*, 2018, pp. 108–115.

[69] T. Byun and S. Rayadurgam, "Manifold for machine learning assurance," in *Proceedings of the International Conference on Software Engineering, New Ideas and Emerging Results*, 2020, pp. 97–100.

[70] D. Gopinath, C. S. Pasareanu, K. Wang, M. Zhang, and S. Khurshid, "Symbolic execution for attribution and attack synthesis in neural networks," pp. 282–283, 2019.

[71] Y. Sun, M. Wu, W. Ruan, X. Huang, M. Kwiatkowska, and D. Kroening, "Concolic testing for deep neural networks," in *Proceedings of the International Conference on Automated Software Engineering*, 2018, pp. 109–119.

[72] H. Zhou, W. Li, Z. Kong, J. Guo, Y. Zhang, B. Yu, L. Zhang, and C. Liu, "Deepbillboard: Systematic physical-world testing of autonomous driving systems," in *Proceedings of the International Conference on Software Engineering*, 2020, pp. 347–358.

[73] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," *CoRR*, vol. abs/1612.06299, 2016.

[74] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black-box safety testing of deep neural networks," in *Proceedings of the Tools and Algorithms for the Construction and Analysis of Systems*, 2018, pp. 408–426.

[75] J. Wang, H. Qiu, Y. Rong, H. Ye, Q. Li, Z. Li, and C. Zhang, "Bet: Black-box efficient testing for convolutional neural networks," in *Proceedings of the International Symposium on Software Testing and Analysis*, 2022, p. 164–175.

[76] C. Wan, S. Liu, S. Xie, Y. Liu, H. Hoffmann, M. Maire, and S. Lu, "Automated testing of software that uses machine learning apis," in *Proceedings of the International Conference on Software Engineering*, 2022, pp. 212–224.