This is an Accepted Manuscript version of the following article, accepted for publication in:

A. Arrieta, "On the Cost-Effectiveness of Composite Metamorphic Relations for Testing Deep Learning Systems," 2022 IEEE/ACM 7th International Workshop on Metamorphic Testing (MET), 2022, pp. 42-47, doi: 10.1145/3524846.3527335.

On the Cost-Effectiveness of Composite Metamorphic Relations for Testing Deep Learning Systems

Aitor Arrieta
Mondragon university
Mondragon, Spain
aarrieta@mondragon.edu

ABSTRACT

Deep Learning (DL) components are increasing their presence in mission and safety-critical systems, such as autonomous vehicles. The verification process of such systems needs to be rigorous, for which automated solutions are paramount. To allow test automation, test oracles are necessary. In the context of DL systems, metamorphic test oracles have found to be effective. However, such oracles require the execution of multiple tests, which makes testing more expensive. Metamorphic relation composition can reduce the cost of metamorphic testing. However, its effectiveness has found mixed answers. This paper reports the preliminary results of our study on measuring the cost-effectiveness of composite metamorphic relations for testing DL systems. To this end, we empirically evaluate the cost-effectiveness of composite metamorphic relations within a DL model for object classification. Our results suggest that composite metamorphic relations reduce the failure revealing capability when compared to their component metamorphic relations.

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**.

KEYWORDS

Metamoprhic Testing, Metamorphic Relation Composition, Deep Learning Systems

ACM Reference Format:
Aitor Arrieta. 2022. On the Cost-Effectiveness of Composite Metamorphic Relations for Testing Deep Learning Systems. In *7th International Workshop on Metamorphic Testing (MET'22), May 9, 2022, Pittsburgh, PA, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3524846.3527335

1 INTRODUCTION

Deep Learning (DL) components are commonly integrated into critical software systems that need to perform complex tasks, including image processing or obstacle detection of autonomous vehicles [22]. Thoroughly testing such systems is paramount to ensure a high dependability of critical systems. Metamorphic oracles, which take advantage of metamorphic relations (MRs) between input values, have emerged as a viable approach to overcome the test oracle problem of DL systems [14]. Metamorphic testing has demonstrated success in alleviating the oracle problem in a wide range of DL applications, including autonomous driving systems [21], image classification [17] and object detection [17].

However, the application of metamorphic testing consumes additional testing resources as it involves multiple program executions [13]. To increase the cost-effectiveness of metamorphic testing, prior studies have proposed to reduce the number of MRs through *MR composition* [6, 7]. Nevertheless, whether the fault detection capability of the composite MR is higher or similar to its corresponding component MRs has been controversial. While Dong et al. [6] reported that the fault detection capability remains unchanged, Liu et al. [7] found that the fault detection capability may be reduced after MR composition.

To explore under what situations the fault detection capability of MR compositions could be reduced, a recent study performed a theoretical and empirical analysis of the effectiveness of MR composition [13]. In their study, Qiu et al. defined a general guide on when a composite MR should be used instead of their component MRs [13]. However, the effectiveness of techniques for testing traditional software systems might be different to that of DL systems. This is mainly due to the fact that a large part of the program logic of DL systems is determined by the data used for training [14].

Subsequently, in this study we perform a preliminary analysis of the cost-effectiveness of composite MRs for testing DL systems, aiming to answer the following research question:

> ***How does the cost-effectiveness of the composite MR compare with that of its corresponding component MRs when testing DL systems?***

Specifically, we applied it to the context of image classification using the Resnet50 Convolutional Neural Network (CNN). We selected a set of MRs that follow the guidelines proposed by Qiu et al., [13] to form several MR compositions and compare their cost-effectiveness against their component MRs. These MRs were selected based on the study by Spieker and Gotlieb for image classification [17]. Our evaluation suggests that in the context of DL systems testing, composite MRs are not recommended. For all the cases, we found that at least one of the individual MRs outperformed the composite MR in terms of missclassification rate (i.e., violation of MRs). To the best of our knowledge, this is the first paper that investigates the cost-effectiveness of composite MRs in the context of DL systems. Our preliminary results suggest that composite MRs do not increase the cost-effectiveness of metamorphic testing in the context of DL systems.

MET'22, May 9, 2022, Pittsburgh, PA, USA

ACM ISBN 978-1-4503-9307-2/22/05…$15.00
https://doi.org/10.1145/3524846.3527335

The rest of the paper is structured as follows: Section 2 explains the background and basic concepts of composite MRs. Section 3 describes the empirical evaluation we have carried out. Section 4 positions our work with the current literature. Section 5 discusses the limitations of our evaluation and identifies a set of challenges. Lastly, Section 6 concludes the paper and discusses future research directions.

2 BACKGROUND AND BASIC CONCEPTS

We now explain the basic notation and background to understand our paper. Most of the formal notations from Section 2.1 are borrowed from the study by Qiu et al., [13]. For further details the reader is referred to that paper.

2.1 MR Composition

The MR Composition was originally proposed as a method to generate new MRs from existing ones [6, 7]. The following example illustrates the basic concept of MR composition [13]. Considering the following MRs corresponding to the properties of a *sine* function:

- MR_1: If $x' = -x$, then $sin(x') = -sin(x)$;
- MR_2: If $x' = x + 2\pi$, then $sin(x') = sin(x)$

it is possible to compose MR_1 and MR_2 together to compose the metamorphic relation MR_{12}, i.e., $MR_1(MR_2)$ [13], which can be expressed as:

- If $x' = -(x + 2\pi)$, then $sin(x') = -sin(x)$

In this case, we call MR_{12} the *Composite MR*, whereas MR_1 and MR_2 are the component MRs of MR_{12}.

Figure 1 shows another example of how to form a composite MR for the context of DL systems aiming to classify images. The MRs follow the intuition that changing the original image shall not affect the classification label returned by the DL model. Based on two MRs defined by Spieker and Gotlieb [17], i.e., flip the original image from left to right (MR_1) and flip the original image from up to down (MR_2), we can compose MR_{12}, which has both the image flipped from left to right and from up to down.

2.1.1 A special class of metamorphic relations (MRs). The specific class of MRs that are considered both in this study and in [13] is defined as follows:

Let

- $f : \tau \rightarrow \mathfrak{R}$ be a targeted function;
- $I : T \rightarrow T'$ (where $T \subseteq \tau$, and $T' = I(T) \subseteq \tau$ be a mapping that takes in a source input and generates a follow-up input for f
- $O : R \rightarrow R'$ (where $R = f(T)$ and $R' = O(R) \subseteq \mathfrak{R}$ be a mapping that takes in a source output (i.e., $f(t)$) and generates a follow-up output.

A metamorphic relation *MR* is a necessary property of f. *MR* is formally expressed as follows [13]:

$$\forall t \in T(f(I(t)) = O(f(t))) \tag{1}$$

where,

- T is the set of source inputs for MR;
- I and O are the *input* and *output mappings* of *MR*;
- t is a *source input* of *MR*, where $t \in T$;
- $I(t)$ is the *follow-up input* corresponding to t;
- $f(t)$ is the *source output* corresponding to t;
- $f(I(t))$ is the *follow-up output* corresponding to t.

This definition of special class of MRs has two assumptions [13]: (1) an MR involves two separate mapping (i.e., the input mapping I and the output mapping O); and (2) the input and output mappings involve a single input and a single output respectively.

2.1.2 Composable MR. We now define what a composable MR is [13].

Let

- $f : \tau \rightarrow \mathfrak{R}$ be a targeted function;
- MR_x and MR_y be two MRs of f

MR_x is composable with MR_y if [13]:

- $I_y(T_y) \subseteq T_x$, that is, the range of I_y is a subset of the domain of I_x;
- $O_y(R_y) \subseteq R_x$ (where $R_x = f(T_x)$ and $R_y = f(T_y)$), i.e., the range of O_y is a subset of the domain of O_x

Notice that the subscripts are used to link an MR and its related components (e.g., T_x, I_x and O_x refer to the set of source inputs, input mapping and output mapping of MR_x).

2.1.3 Composite MR and Component MR. We now define the construction of a composite MR [13]:

Let

- $f : \tau \rightarrow \mathfrak{R}$ be a targeted function;
- MR_x and MR_y be two MRs of f
- MR_x be composable with MR_y

The composite MR (i.e., MR_{xy}), composed by MR_x and MR_y is formally expressed as follows:

$$\forall t \in T_{xy}(f(I_{xy}(t)) = O_{xy}(f(t))), \tag{2}$$

where

- $T_{xy} = T_y$;
- $I_{xy}(t) = I_x(I_y(t))$;
- $O_{xy}(f(t)) = O_x(O_y(f(t)))$.

2.2 Guidelines for the use of composite MR

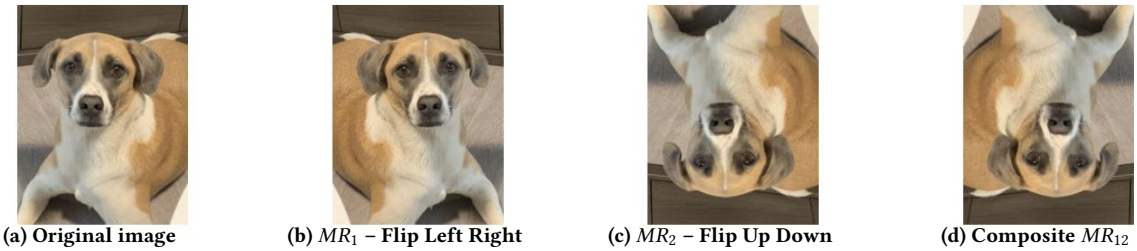
Based on a solid theoretical and empirical analysis, Qiu et al. [13] concluded that the composite MR (i.e., MR_{xy}) should be used instead their corresponding component MRs (i.e., MR_x and MR_y) if:

- Both MR_x and MR_y belong to the special class of MRs in accordance with the definition of MR provided in Section 2.1.1;
- MR_x is composable with MR_y according to the definition from Section 2.1.2;
- $I_y(T_y) = T_x$, I_y is bijective, and O_x is injective.

In our experiments, our MRs follow such characteristics.

2.3 Deep Learning Systems

Deep Learning (DL) is the associated learning technique of Neural Networks (NN), one type of machine-learning algorithm that has



(a) Original image **(b) MR_1 – Flip Left Right** **(c) MR_2 – Flip Up Down** **(d) Composite MR_{12}**

Figure 1: Examples of two MRs applied to an image and their corresponding composite MR

gained most attention [14]. These systems offer statistical techniques to learn complex patterns from training data [14]. The learned functions can later be applied in unseen data, allowing to make predictions about unknown properties of observed data [14]. When learned, these functions can be represented and stored as a set of hyper-parameters and variables in a *model*, which is defined as an instance of a specific machine-learning algorithm [14].

3 EMPIRICAL EVALUATION

The research question (RQ) that our evaluation is aiming to answer is the following:

How does the cost-effectiveness of the composite MR compare with that of its corresponding component MRs when testing DL systems?

While the cost-effectiveness of composite MRs have been investigated [13], to the best of our knowledge, this is the first evaluation targeting DL systems. To this end, we analyze the cost-effectiveness of composite MRs by proposing an empirical evaluation, which is explained in the following section.

3.1 Experimental setup

To answer this question we used a pre-trained image classification network (the Resnet50). We used a total of four MRs (one of them configured to have two sub instances), which were previously defined by Spieker and Gotlieb for image classification purposes [17] and formed a total of eight composite MRs. We measured both the cost in terms of time required to execute the entire test suite, and the effectiveness in terms of the failure detection rate.

3.1.1 Deep learning model. To answer our RQ, we used the pre-trained ResNet50 DL model. This model is a Convolutional Neural Network (CNN) of 50 layers, and is intended for object classification. Specifically, this DL model can classify images into 1,000 object categories and their image input size is 224-by-224.

3.1.2 Dataset and used test suites. We used the Imagenette dataset,[1] which is a subset of 10 classified classes from Imagenet [5], a widely used dataset for image classification algorithms. Spieker and Gotlieb [17] used the CIFAR-10 dataset. Nevertheless, we did not use that dataset because their authors removed it due to a variety of reasons and asked the community not to use it.[2]

[1]https://github.com/fastai/imagenette

[2]See the following page for the explanations given by the CIFAR-10 authors: http://groups.csail.mit.edu/vision/TinyImages/

Similar to Spieker and Gotlieb [17], we divided the dataset into multiple test suites. These test suites are based on the different categories of the images of Imagenette's validation pool. Therefore, we formed a total of six different test suites with different images each and analyzed the cost-effectiveness of the different MRs under different test suites. Table 1 summarizes the main characteristics of the used test suites.

Table 1: Summary of the the test cases used in our study

Test Suite	# of Images
Gas Pump	806
Tench	750
Tape Player	706
Chain Saw	766
Church	788
French Horn	754

3.1.3 Defined Metamorphic Relations. We selected four different MRs [17], one of which (rotation of an image) was instantiated into two MRs. Specifically, we selected the MRs "flip left-right", "flip up-down", "rotate image" and "shear". Two MRs were configurable: rotate image and shear. To comply with the guidelines proposed by Qiu et al. [13] to compose MRs, the configuration was fixed. For the rotation MR, we divided the rotation into two different MRs. One of them rotated the image 5°, whereas the other one rotated the image -5°. We first started with rotations of 30°, but we found that this led to constant missclassifications of the images by the DL models. Therefore, we reduced the rotation angle to 5°. On the other hand, the sheer MR also had a configuration variable to specify the sheer degree. We fixed it to one predefined value to maintain the guidelines [13]. With the five instances of MRs, we formed a total of 8 MR compositions and assessed their cost-effectiveness. It is important to reiterate that the composed composite MRs follow all the three points mentioned in the guide performed by Qiu et al. [13].

Spieker and Gotlieb also suggested three additional MRs [17]: (1) blurring the image, (2) inverting the image and (3) converting the image into black and white. For the first MR, the implementation we found was too slow, reducing significantly the amount of experiments we could afford. For the second MR, the algorithm could not perform right predictions. For the last MR, the black and white format was not accepted by the DL model. Therefore, these three MRs were discarded.

Table 2: Summary of the experimental results

			Effectiveness				Cost (seconds)		
Test Suite	MR_x	MR_y	**FR** MR_x	**FR** MR_y	**FR** $MR_x \cup MR_y$	**FR** MR_{xy}	**Time** MR_x	**Time** MR_y	**Time** MR_{xy}
Gas Pump	flip left right	flip up down	0.122	0.474	0.489	0.242	34.914	35.151	35.241
	flip left right	rotate - 5°	0.122	0.233	0.275	0.124	34.894	35.333	35.450
	flip left right	rotate 5°	0.122	0.218	0.261	0.110	35.469	36.079	36.247
	flip left right	shear	0.122	0.208	0.256	0.104	35.536	47.105	47.674
	flip up down	rotate - 5°	0.474	0.233	0.531	0.248	36.039	35.883	36.768
	flip up down	rotate 5°	0.474	0.218	0.511	0.241	36.812	37.151	37.487
	flip up down	shear	0.474	0.208	0.516	0.254	35.666	46.851	46.596
	rotate - 5°	shear	0.233	0.208	0.303	0.119	35.952	48.026	48.389
Tench	flip left right	flip up down	0.112	0.379	0.395	0.188	32.146	32.478	32.594
	flip left right	rotate - 5°	0.112	0.235	0.253	0.121	32.061	32.232	32.321
	flip left right	rotate 5°	0.112	0.211	0.232	0.115	31.905	32.506	32.813
	flip left right	shear	0.112	0.144	0.187	0.073	31.838	41.088	40.857
	flip up down	rotate - 5°	0.379	0.235	0.445	0.293	32.670	32.681	32.950
	flip up down	rotate 5°	0.379	0.211	0.429	0.304	32.716	32.750	33.315
	flip up down	shear	0.379	0.144	0.408	0.235	32.298	41.047	41.916
	rotate - 5°	shear	0.235	0.144	0.256	0.147	29.036	34.670	34.799
Tape Player	flip left right	flip up down	0.204	0.467	0.530	0.249	31.959	32.347	32.183
	flip left right	rotate - 5°	0.204	0.368	0.442	0.204	30.354	30.842	30.887
	flip left right	rotate 5°	0.204	0.391	0.467	0.194	30.384	31.052	30.909
	flip left right	shear	0.204	0.306	0.399	0.132	30.521	39.283	39.328
	flip up down	rotate - 5°	0.467	0.368	0.646	0.305	30.645	30.665	30.927
	flip up down	rotate 5°	0.467	0.391	0.620	0.265	30.768	30.911	31.095
	flip up down	shear	0.467	0.306	0.584	0.275	28.479	33.722	33.800
	rotate - 5°	shear	0.368	0.306	0.487	0.222	27.425	32.421	32.733
Chain Saw	flip left right	flip up down	0.178	0.480	0.507	0.228	29.246	29.262	29.427
	flip left right	rotate - 5°	0.178	0.347	0.392	0.192	29.934	30.014	29.890
	flip left right	rotate 5°	0.178	0.366	0.394	0.184	29.713	29.795	30.176
	flip left right	shear	0.178	0.292	0.339	0.129	29.866	35.624	35.506
	flip up down	rotate - 5°	0.480	0.347	0.559	0.303	29.852	29.824	30.278
	flip up down	rotate 5°	0.480	0.366	0.546	0.292	30.888	31.343	31.096
	flip up down	shear	0.480	0.292	0.527	0.273	31.377	40.457	40.897
	rotate - 5°	shear	0.347	0.292	0.415	0.218	31.776	41.616	41.562
Church	flip left right	flip up down	0.165	0.881	0.904	0.439	33.966	34.703	34.597
	flip left right	rotate - 5°	0.165	0.414	0.449	0.190	34.088	34.674	34.694
	flip left right	rotate 5°	0.165	0.348	0.404	0.179	34.033	34.727	34.722
	flip left right	shear	0.165	0.287	0.343	0.145	34.127	44.117	43.762
	flip up down	rotate - 5°	0.881	0.414	0.919	0.464	34.217	34.606	34.943
	flip up down	rotate 5°	0.881	0.348	0.914	0.473	34.645	34.529	35.114
	flip up down	shear	0.881	0.287	0.901	0.445	34.505	43.683	45.176
	rotate - 5°	shear	0.414	0.287	0.487	0.223	34.591	43.959	44.914
French Horn	flip left right	flip up down	0.133	0.379	0.408	0.196	29.766	29.822	29.958
	flip left right	rotate - 5°	0.133	0.223	0.271	0.109	32.405	33.212	32.921
	flip left right	rotate 5°	0.133	0.218	0.260	0.129	33.702	33.616	33.575
	flip left right	shear	0.133	0.236	0.284	0.098	32.952	44.340	44.284
	flip up down	rotate - 5°	0.379	0.223	0.422	0.237	32.193	32.625	32.677
	flip up down	rotate 5°	0.379	0.218	0.419	0.232	32.370	33.018	32.757
	flip up down	shear	0.379	0.236	0.440	0.206	32.676	43.849	44.533
	rotate - 5°	shear	0.223	0.236	0.321	0.154	31.275	40.545	41.946

3.1.4 Evaluation metrics. The **effectiveness** of the techniques was measured in terms of the failure rate (FR) of the component MRs (both alone and combined) as well as the composite MR. FR image misclassification rate (i.e., if the test suite has 30 images and 5 are misclassified, FR will be 5/30). As in a previous study [17], the failure rate was considered based on the total amount of missclassification detected by the MRs.

The **cost** of the techniques was measured in terms of the entire test suite test execution time, which also considered the generation of the follow-up test cases based on the defined MRs. This was considered because there might be cases where generating the follow-up test case takes a long time (e.g., in the case of the blurring MR, which finally was not selected, it took a long time to generate the follow-up test case). In addition, since the execution time is not deterministic, we repeated the execution of tests 10 times, to account for the random variations. All the reported time values are the average of these 10 runs.

3.2 Analysis of the Results and Discussion

Table 2 reports the results of the performed evaluation. Columns FR MR_x and FR MR_y report the Failure Rate (FR) of MR_x and MR_y respectively. Column FR $MR_x \cup MR_y$ reports the cumulative FR of both MRs. Lastly, column FR MR_{xy} reports the FR obtained by the composite MR of MR_x and MR_y. As can be seen, for all the cases, the FR of the composite MR (i.e., MR_{xy}) was lower than the cumulative FR of their corresponding component MRs. Furthermore, in all cases, at least one of the component MRs of the composite MR had a higher failure rate. In addition, in 20 out of a total of 48 cases, both component MRs obtained individually a higher effectiveness than their composite MR. The reduction in the failure rate is very large in most cases. For instance, there was a decrease of more than 50% in the failure rate of the composite MR with respect to the cumulative failure rate of their corresponding component MRs in 36 out of 48 cases.

As for the cost, it can be appreciated that the time required for the composite MR is similar to the ones related to individual MRs. This means that most of the overhead is produced by the execution of the test (i.e., in this case, the inference of the DL model for classifying an image). Conversely, the generation of the follow-up test case seems to be negligible in most cases. It can be appreciated that in those cases where the shear MR exists, the time for executing the test suite slightly increases. This could be because the generation

of the follow-up test case might be computationally heavier than for the rest of MRs. It is noteworthy that in this case, the time for executing a test is relatively fast because we are testing the DL component alone. However, when such system is integrated with the rest of the system, the execution time can be significantly increased. This can be especially exacerbated in context like Cyber-Physical Systems, where DL models are integrated with physical components modeled through complex mathematical models (e.g., autonomous driving systems [2, 18]).

The current results suggest that while composite MRs reduce the execution time by around half of the time when compared to executing both component MRs, the failure detection capability of them is reduced in the context of DL systems. In contrast, at least one of the component MRs showed a higher failure rate. Therefore, based on our preliminary empirical evaluation, we can respond to the RQ of our study as follows:

Composite MRs do not increase the cost-effectiveness of metamorphic testing of DL systems when compared to their component MRs. Specifically, our evaluation suggests that the effectiveness is reduced.

3.3 Threats to Validity

We now summarize the threats to validity of our study and how we tried to mitigate such threats.

An *internal validity* threat in our evaluation could be related to the selected MRs. To reduce this threat, we selected these MRs based on a previous work on MT of DL systems for image classification [17]. Some of the selected MRs have parameters (e.g., degrees to be rotated). To make our MRs compliant with the guidelines for composing MRs [13], these parameters had to be fixed. We fixed them based on some manually carried out preliminary evaluations. For instance, when setting the degrees to be rotated to 30º, most of the predictions by the DL model were labeled as "PC monitor", due to the edges that this MR provokes in the follow-up test cases. Therefore, we reduced such parameter values. Other values of these parameters could have led to other results. However, we believe that the selected values are appropriate and reasonable for our evaluation.

An *external validity* threat in our study relates to the generalization of the results. We only applied the different composite MRs to one single DL model. However, the precision of such DL model is one of the highest in the state-of-the-art for the classification of images. Furthermore, to reduce this threat, we evaluated the different approaches with different test suites.

A *conclusion validity* threat in our evaluation is related to the cost (i.e., execution time) of the different methods. We considered this by measuring the time required by the different methods to execute the entire test suite. This time is non-deterministic. Therefore, we executed each instance of comopsite and component MRs 10 times and reported the average values of the times.

4 RELATED WORK

In the last few years, research on testing DL and ML systems has significantly increased, as can be seen in recent surveys and systematic mapping studies [14, 20]. This research spans across different areas [14, 20], including research on adapting and proposing new test adequacy criteria for DL systems testing [3, 4, 8–10, 12], test generation of DL systems [8, 15], comparison between on-line and off-line DL system testing [2], etc.

The survey by Zhangh et al., identified the test oracle problem as one of the fundamental challenges when testing ML systems [20]. Therefore, much of the DL testing literature seeks to find techniques tackling the test oracle problem [14, 20]. In such a context, metamorphic oracles have been found the most widely applied technique [14]. For instance, Xie et al. [19] investigated the application of metamorphic testing to test unsupervised ML systems. Ding et al. proposed an approach for validating the classification accuracy of a DL framework that included a CNN, a DL execution environment, and a massive image data set. Murphy et al. [11] proposed metamorphic testing to test different ML algorithms (e.g., Support Vector Machines). Saha and Kanewala compared multiple MRs from previous studies to test supervised classifiers [16]. Similar to this work, Dwarakanath et al., proposed a set of MRs, which included rotation of images, to test image classifiers [1]. In contrast to all these studies, which consider MR individually, our work aims at analyzing the cost-effectiveness of composite MRs.

To the best of our knowledge, the cost-effectiveness of MRs are studied only in three works [6, 7, 13]. Qiu et al. [13] used an ML classifier algorithm based on the K-Nearest Neighbors (KNN) as one of the four studied cases. However, this algorithm is different to DL systems, which are based on Deep Neural Networks. Thus, in contrast to these studies, our targeted systems are DL systems. To the best of our knowledge, this is the first study that investigates the cost-effectiveness of composite MRs in such a context.

5 LIMITATIONS AND IDENTIFIED CHALLENGES

This paper presents a preliminary evaluation that studies the cost-effectiveness of composite MRs when testing DL systems. Our study suggests that composite MRs reduce the failure detection when compared to their component MRs. However, our study may also suffer certain limitations, which can be summarized in the following points:

Limitation 1 – Application context: We only evaluated composite MRs in the context of DL systems for image classification. Nevertheless, the application of DL systems spans several areas, including natural language processing, autonomous driving systems, object identification, etc. In the future, more research is required to assess (1) whether composite MRs are appropriate in other DL domains and (2) under which circumstances MR composition may increase the cost-effectiveness in DL systems testing.

Limitation 2 – Empirical evaluation: Our evaluation has been limited to empirical findings. Changes in the empirical set-up (e.g., change a dataset) may result in different conclusions of the study. Ideally, a theoretical study should be used to complement and explain our empirical findings.

Limitation 3 – Applied to off-line DL testing: Our evaluation has been carried out in the context of the so-called "off-line" testing of the DL component [2]. However, the so-called "on-line" testing takes longer time to execute the test suites. We believe that composite MRs may have greater potential in such cases where the difference in execution times can be huge.

Based on these limitations, we believe that future work could be centered on solving different challenges:

Challenge 1 – To provide new guidelines for composite MRs in the context of DL systems: Our preliminary findings suggest that composite MRs might not be cost-effective for DL systems testing. We believe that guidelines specific to composite MRs are necessary to understand when an MR composition should be used in the context of DL systems. While empirical evidence might have its limitations, a theoretical analysis might be challenging to be applicable in a context like DL systems, where the functionality the DL system is largely driven by the training data.

Challenge 2 – Alternatives to MR composition: In the event that we discover that composite MRs are not applicable in the context of DL systems, alternatives to such technique might need to be investigated. Traditional regression test selection and prioritization techniques could be adapted as an alternative to MR composition to maximize the cost-effectiveness of MT in the context of DL systems. In such cases, adequacy criteria (e.g., surprise adequacy [4]) could be used, for instance, to prioritize both test cases and MRs in the context of regression testing of DL systems in an MLOps pipeline.

6 CONCLUSION AND FUTURE WORK

In this short-paper we report our preliminary results on the cost-effectiveness of composite MRs when testing DL systems. Prior studies have shown that such techniques can increase the cost-effectiveness of metamorphic testing for traditional software systems [13]. However, since the functionality of DL systems is driven by the data used for training them, the guidelines proposed by Qiu et al. may need to be revisited and adapted for such context [13]. To this end, we used a DL model for image classification, adapted four MRs from an existing work [4] and formed a total of eight composite MRs. Our initial findings suggest that composite MRs reduce the failure revealing capability of their component MRs, therefore, are not applicable for the context of DL testing. Nevertheless, our study is preliminary and has a set of limitations that have been identified.

In the close future we would like to continue this study from different perspectives. Firstly, we would like to analyze the cost-effectiveness of composite MRs with other DL models. Secondly, we would like to analyze other DL application contexts, such as object recognition and prediction of the steering angle of an autonomous vehicle. Lastly, we would like to extract conclusions to develop guidelines for the application of composite MRs in the context of DL systems.

REPLICATION PACKAGE

The replication package of the paper can be found in Zenodo: https://doi.org/10.1145/3524846.3527335

ACKNOWLEDGMENTS

This publication is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871319.

REFERENCES

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

[2] Fitash Ul Haq, Donghwan Shin, Shiva Nejati, and Lionel C Briand. 2020. Comparing offline and online testing of deep neural networks: An autonomous car case study. In *2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST)*. IEEE, 85–95.

[3] Nargiz Humbatova, Gunel Jahangirova, and Paolo Tonella. 2021. Deepcrime: Mutation testing of deep learning systems based on real faults. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 67–78.

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

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

[6] Dong Guowei Xu Baowen Chen Lin and Nie Changhai Wang Lulu. 2008. Case studies on testing with compositional metamorphic relations. *Journal of Southeast University (English Edition)* 4 (2008).

[7] Huai Liu, Xuan Liu, and Tsong Yueh Chen. 2012. A new method for constructing metamorphic relations. In *2012 12th International Conference on Quality Software*. IEEE, 59–68.

[8] Lei Ma, Felix Juefei-Xu, Minhui Xue, Bo Li, Li Li, Yang Liu, and Jianjun Zhao. 2019. Deepct: Tomographic combinatorial testing for deep learning systems. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 614–618.

[9] Lei Ma, Felix Juefei-Xu, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Chunyang Chen, Ting Su, Li Li, Yang Liu, et al. 2018. Deepgauge: Multi-granularity testing criteria for deep learning systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*. 120–131.

[10] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, et al. 2018. Deepmutation: Mutation testing of deep learning systems. In *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 100–111.

[11] Christian Murphy, Kuang Shen, and Gail Kaiser. 2009. Automatic system testing of programs without test oracles. In *Proceedings of the eighteenth international symposium on Software testing and analysis*. 189–200.

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

[13] Kun Qiu, Zheng Zheng, Tsong Chen, and Pak-Lok Poon. 2020. Theoretical and Empirical Analyses of the Effectiveness of Metamorphic Relation Composition. *IEEE Transactions on Software Engineering* (2020).

[14] Vincenzo Riccio, Gunel Jahangirova, Andrea Stocco, Nargiz Humbatova, Michael Weiss, and Paolo Tonella. 2020. Testing machine learning based systems: a systematic mapping. *Empirical Software Engineering* 25, 6 (2020), 5193–5254.

[15] Vincenzo Riccio and Paolo Tonella. 2020. Model-based exploration of the frontier of behaviours for deep learning system testing. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 876–888.

[16] Prashanta Saha and Upulee Kanewala. 2019. Fault detection effectiveness of metamorphic relations developed for testing supervised classifiers. In *2019 IEEE International conference on artificial intelligence testing (AITest)*. IEEE, 157–164.

[17] Helge Spieker and Arnaud Gotlieb. 2020. Adaptive metamorphic testing with contextual bandits. *Journal of Systems and Software* 165 (2020), 110574.

[18] Andrea Stocco, Michael Weiss, Marco Calzana, and Paolo Tonella. 2020. Misbehaviour prediction for autonomous driving systems. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*. 359–371.

[19] Xiaoyuan Xie, Zhiyi Zhang, Tsong Yueh Chen, Yang Liu, Pak-Lok Poon, and Baowen Xu. 2020. METTLE: a METamorphic testing approach to assessing and validating unsupervised machine LEarning systems. *IEEE Transactions on Reliability* 69, 4 (2020), 1293–1322.

[20] Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering* (2020).

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

[22] Tahereh Zohdinasab, Vincenzo Riccio, Alessio Gambi, and Paolo Tonella. 2021. Deephyperion: exploring the feature space of deep learning-based systems through illumination search. In *Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*. 79–90.