The Journal of Systems & Software 182 (2021) 111060

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

# Using metamorphic relations to verify and enhance Artcode classification[☆,☆☆]

Liming Xu[a], Dave Towey[b,*], Andrew P. French[c], Steve Benford[c], Zhi Quan Zhou[d], Tsong Yueh Chen[e]

[a] Department of Engineering, University of Cambridge, Cambridge, United Kingdom
[b] School of Computer Science, University of Nottingham Ningbo China, Ningbo, China
[c] School of Computer Science, University of Nottingham, Nottingham, United Kingdom
[d] School of Computing and Information Technology, University of Wollongong, Australia
[e] Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia

ARTICLE INFO

*Article history:*
Received 1 December 2020
Received in revised form 18 June 2021
Accepted 27 July 2021
Available online 18 August 2021

*Keywords:*
Metamorphic testing
Metamorphic relation
Classification
Software verification
Machine learning
Artcode

ABSTRACT

Software testing is often hindered where it is impossible or impractical to determine the correctness of the behaviour or output of the software under test (SUT), a situation known as the *oracle problem*. An example of an area facing the oracle problem is automatic image classification, using machine learning to classify an input image as one of a set of predefined classes. An approach to software testing that alleviates the oracle problem is metamorphic testing (MT). While traditional software testing examines the correctness of individual test cases, MT instead examines the relations amongst multiple executions of test cases and their outputs. These relations are called metamorphic relations (MRs): if an MR is found to be violated, then a fault must exist in the SUT. This paper examines the problem of classifying images containing visually hidden markers called *Artcodes*, and applies MT to verify and enhance the trained classifiers. This paper further examines two MRs, Separation and Occlusion, and reports on their capability in verifying the image classification using one-way analysis of variance (ANOVA) in conjunction with three other statistical analysis methods: t-test (for unequal variances), Kruskal–Wallis test, and Dunnett's test. In addition to our previously-studied classifier, that used Random Forests, we introduce a new classifier that uses a support vector machine, and present its MR-augmented version. Experimental evaluations across a number of performance metrics show that the augmented classifiers can achieve better performance than non-augmented classifiers. This paper also analyses how the enhanced performance is obtained.

## 1. Introduction

Over the past two decades, machine learning techniques have been widely adopted by research communities (e.g., computer vision, bioinformatics, computational linguistics, and medical imaging) to solve a range of practical problems. For researchers in the machine learning and software testing communities, the ability to build accurate learning models and verify their quality is essential. Due to the nature of machine learning programmes, test oracles (mechanisms to categorically determine if the software behaviour or output is correct) are generally very difficult to define. Hence, conventional software testing techniques may not be effective for detecting defects. The issue of how to ensure the quality of applications based on machine learning has become increasingly important (Xie et al., 2011).

*Metamorphic testing* (MT) is a testing technique that can alleviate the *oracle problem* (Chen et al., 1998, 2003), a major challenge in software testing. While conventional testing methods focus on verifying individual outputs, MT examines *relations* among the inputs and outputs of multiple executions of the software under test (SUT). These *relations* are called *metamorphic relations* (MRs). Since the first MT paper (Chen et al., 1998) was published in 1998, MT has been widely used to test software in various fields, including: scientific computing (Ding et al., 2016), numerical analysis (Chen et al., 2002), classification (Xie et al., 2011, 2009), cybersecurity (Chen et al., 2016), image processing (Mayer and Guderlei, 2006), compilers (Le et al., 2014; Donaldson et al., 2017), search engines (Zhou et al., 2016), web security (Mai et al.,

[☆] Editor: Raffaela Mirandola.

[☆☆] This work is an extension of an MET'18 paper (Xu et al., 2018), which was completed while the first author was a Ph.D. student at University of Nottingham Ningbo China.

[*] Corresponding author.
*E-mail addresses:* lx249@cam.ac.uk (L. Xu), dave.towey@nottingham.edu.cn (D. Towey), andrew.p.french@nottingham.edu.cn (A.P. French), steve.benford@nottingham.edu.cn (S. Benford), zhiquan@uow.edu.au (Z.Q. Zhou), tychen@swin.edu.cn (T.Y. Chen).

https://doi.org/10.1016/j.jss.2021.111060
0164-1212/© 2021 Elsevier Inc. All rights reserved.

2020), and visualisation (McNutt et al., 2020), among others. A body of literature also describes its integration with other testing techniques to improve their applicability and effectiveness. Comprehensive surveys about MT have also been recently published by Segura et al. (2016) and Chen et al. (2018).

More recently, MT has been increasingly gaining interest in classic AI fields for testing systems powered by machine learning, including: machine translation (Zhou and Sun, 2018; He et al., 2019), autonomous driving (Zhang et al., 2018; Zhou and Sun, 2019), and generic NLP (natural language processing) models (Ma et al., 2020; Ribeiro et al., 2020). MT can have comparable bug-revealing effectiveness to model-based testing, and hence is a useful alternative to test an implementation, especially in situations where a model is expensive to construct (Hughes, 2020).

MT techniques have been used to test machine learning programmes (Xie et al., 2011, 2009; Murphy et al., 2008). Machine learning techniques have also been used to automatically identify MRs, although so far only with simple MRs (Kanewala and Bieman, 2013). Xu et al. (2018) expanded the traditional role of MRs from software testing to a kind of *post adjustor* for a machine learning programme, building a more accurate learning model using an example of the Artcode classification problem. Artcodes are visual codes whereby bespoke designs can be scanned to trigger the digital information attached to them. Artcodes may be disguised as normal images in the scene through their freeforms and complex aesthetic patterns — and they may appear as any instances of semantic objects. Therefore, it is not straightforward for people to build *scan affordance* without the support of an alert system that can recognise the presence of Artcodes in the context. The core part of such an alert system is Artcode classification, which determines whether or not the Artcode-based augmented reality applications can work effectively. We will present Artcode basics and Artcode classification in more detail in Section 2.2. More information about Artcode applications in augmented reality can be found in the literature, such as Meese et al. (2013), Xu et al. (2017), Benford et al. (2018), and Koleva et al. (2020).

Two MRs, *Separation* and *Occlusion*, identified based on the category of the inputs, were introduced by Xu et al. (2018), who reported on their ability to improve the performance of the original classifier. Initial experimental evaluations showed that MRs could enhance the performance in this case of supervised Artcode classification.

In this paper, we further explore the Separation and Occlusion MRs, present more detailed experimental analyses, and generalise the ability of MRs in both verification and enhancement. Experiments were conducted to show not only the applicability of MT in verifying the correctness of the classifier, but also the improved performance obtained by the MR-augmented framework regardless of the chosen classification methods. The new contributions of this paper are mainly threefold:

(1) We report on the capability of the two MRs to verify the correctness of the previously introduced classification model (Xu et al., 2017) using a set of complementary statistical test methods.
(2) We analyse and discuss how the improved performance of the MR-augmented classifiers is achieved, explaining how the post adjustor rectifies incorrect predictions.
(3) We introduce the use of a Support Vector Machine (SVM) as the classification algorithm in the original classifier and investigate its impact on the performance of the MR-augmented framework, comparing its performance with the MR-augmented classifier based on Random Forests (RF).

The rest of this paper is organised as follows. Section 2 gives a brief description of metamorphic testing and Artcode classification. Section 3 presents the MR-augmented classification framework. The experimental studies examining the MRs' verification and enhancement capability are given in Section 4. Section 5 analyses how the improved performance is obtained by MR augmentation. Finally, Section 6 concludes the paper, highlighting some areas for future work.

## 2. Preliminaries

### 2.1. Metamorphic testing

In software testing, a mechanism that can determine whether a test has passed or failed is called an *oracle*. A situation where the oracle is not available, or is too expensive to be used, is known as the oracle problem (Barr et al., 2015). Metamorphic testing alleviates the oracle problem (Chen et al., 1998). It has been widely adopted in both academia and industry (Chen et al., 2003; Liu et al., 2014; Lindvall et al., 2015; Segura et al., 2016; Zhou et al., 2016; Donaldson et al., 2017; Zhang et al., 2018; He et al., 2019; Mai et al., 2020). MT has successfully detected defects in mature software, including in extensively tested systems (Chen et al., 2015). A central part of MT is a set of MRs, which are relations among several related inputs and their corresponding outputs. While conventional testing approaches uncover software problems by examining the outcome of an individual input, MT detects the presence of a fault by cross-checking multiple related inputs and outputs with respect to MRs.

We next use a database management system (DBMS) example to illustrate the idea of MT. Given two DBMS queries, such as the following:

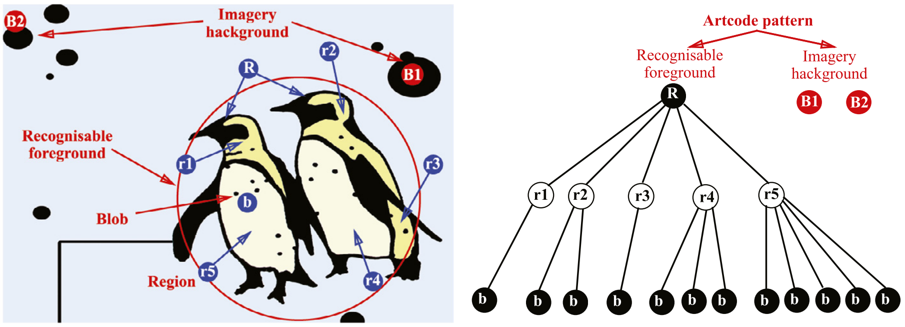
Q1: *select* $*$ *from* student *where* condition_A *and* condition_B;
Q2: *select* $*$ *from* student *where* condition_B *and* condition_A;

the DBMS should return the same results — the outcome for a query with search conditions "A" and "B" and the query that swaps their order should be the same (which could represent an MR). Specifically, if the DBMS returns different results for the queries Q1 and Q2, then a fault must exist in the DBMS implementation.

As with all software testing, MT can only be used to check for the presence of bugs, not their absence (Dijkstra, 1972). For example, a *faulty* DBMS implementation may somehow return the same results for queries Q1 and Q2: thus, although violation of an MR means there must be some fault in the implementation, satisfaction of MRs cannot be taken to mean that the software is fault-free. A key step in MT is the identification of appropriate MRs, which normally requires a good understanding of the problem domain.

### 2.2. Artcode basics and classification

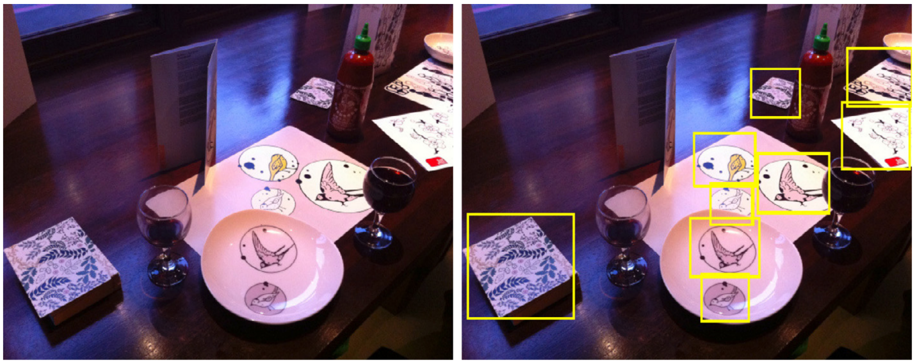
Artcodes are human-designable topological visual markers that are both machine readable and meaningful to humans (Meese et al., 2013). As illustrated in Fig. 1(a), a valid Artcode includes two parts: a recognisable foreground; and some background imagery. The recognisable foreground (the penguins annotated by the red circle) is a closed boundary that is split into several regions (usually five regions, annotated `r1` to `r5` in Fig. 1(a)), with each region containing one or more *blobs* — solid objects disconnected from the region edge. The numbers of these blobs in each region are sorted and joined with a separator to form a string of numbers, which can then be used to represent the Artcode. For example, the code for Fig. 1(a) is "1-1-2-3-5", indicating that there are 1, 1, 2, 3, and 5 blobs found in the respective



(a) Artcode components illustration.

(b) Region adjacency tree.

**Fig. 1.** Illustration of the components of an Artcode (code: "1-1-2-3-5") and the region adjacency tree of its recognisable foreground.



(a) Artcodes-decorated dining context

(b) Artcode examples detected

**Fig. 2.** Illustration of Artcode detection.

regions. Additionally, background imagery (B1 and B2 in Fig. 1(a)) can be added, surrounding the recognisable foreground of an Artcode to enhance the aesthetics, but only if the background does not break the Artcode's topological structure (Costanza and Huang, 2009; Meese et al., 2013). For example, the black solid blobs around the penguins were *intentionally* added to enhance the beauty, but are unconnected to the actual code.

The actual code of an Artcode is represented by a region adjacency tree (RAT) (Costanza and Robinson, 2003); the RATs of the recognisable part of the penguin Artcode and the two background elements are shown in Fig. 1(b). According to the Artcode system, the components are the sets of pixels that are connected to each other, and are known as *connected components*. These connected components are referred to as: *root boundary*, *region*, *blob*, and *background imagery*, depending on their use in the Artcode's context. The root boundary (R) contains several holes (regions) with each having a number of connected components without holes. The number of components is determined by the containment relationship rather than geometrical shapes, as shown in Fig. 1. The components can be any shapes, and this freeform property – with little restriction on shapes – can be an opportunity for designers to create aesthetic, interactive graphics. This property allows Artcode objects to look like an instance of any semantic object classes — animals, flowers, and fish can be recognised as Artcodes if they are designed according to Artcode drawing rules (see Fig. 6).

*Redundancy* is allowed in Artcode design — multiple Artcodes with the same topology but different geometry can appear in an Artcode. Artcodes have been explored in a wide range of contexts (Meese et al., 2013; Benford et al., 2015a,b; Ng and Shaikh, 2016; Thorn et al., 2016; Benford et al., 2016; Preston et al., 2017; Benford et al., 2018; Koleva et al., 2020) since Costanza and Huang (2009) first proposed D-touch markers, whose drawing rules the Artcode system implements and extends.

*Artcode classification.* Fig. 2(a) shows Artcodes being used to augment a dining context, in which the surfaces of objects (menu, plate, and mat) are decorated with Artcodes. In order to alert people to the presence of Artcodes before triggering their further decoding, the first step is to determine whether or not an input image or an image patch contains Artcodes (see Fig. 2(b)). This step involves classification which determines whether an input image is an Artcode or not. This task of *Artcode classification.*[1] involves classifying an input image as either containing an Artcode or not, labelled *Artcode* or *non-Artcode* There is, visually, no obvious difference in appearance or geometrical shape between the two classes (see the examples in Figs. 5 and 6). The geometrical freeform property differentiates Artcodes from other well-known markers, such as barcodes (Woodland and Bernard, 1952), QR codes (International Organization for Standardization, 2015), ARTags (Fiala, 2005), or RUNE-tags (Bergamasco et al., 2011). Artcodes, as a type of augmented reality technique, have been adopted in many situations (as described in Section 2.2) to augment the meanings of the objects in aesthetic-centred contexts. The triggering of the digital information depends on whether or not the presence of Artcodes in the scene is recognised; therefore, Artcode classification is vital for the correct use of Artcode applications and can provide guidelines for other visual codes-based augmented reality techniques. More information about Artcode basics and classification can be found in work such as Costanza and Huang (2009) and Xu et al. (2017).

## 3. MR-augmented classification framework

Conventional classification typically involves two steps: first, create feature vectors that *distinctively* represent each class; and,

[1] *Artcode classification* and *Artcode detection* are used interchangeably in this paper.

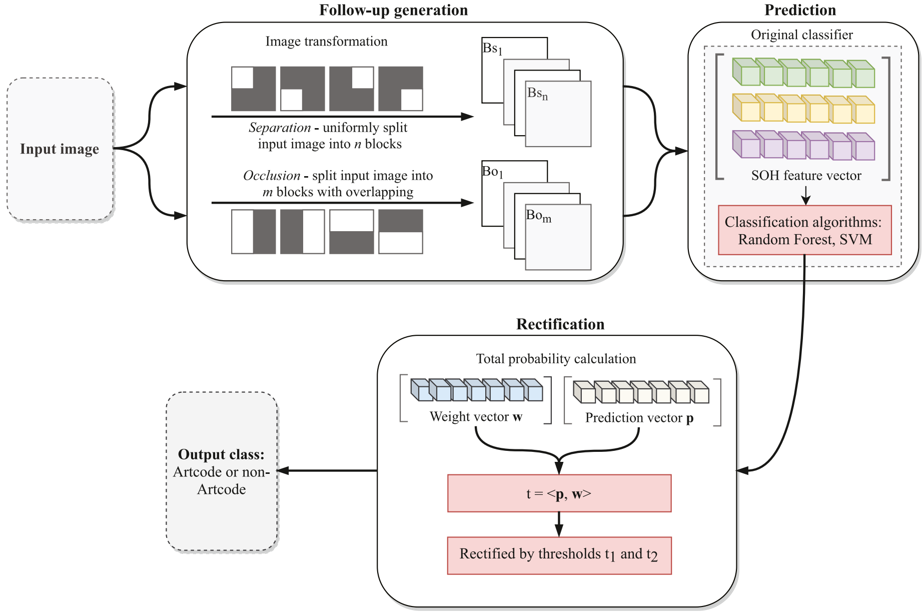
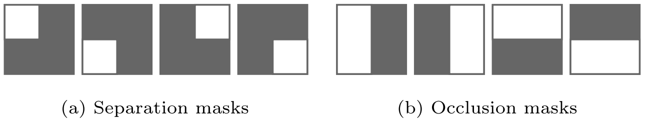

Follow-up generation
Image transformation
Input image
Separation - uniformly split input image into n blocks
Occlusion - split input image into m blocks with overlapping
$Bs_1$
$Bs_n$
$Bo_1$
$Bo_m$
Prediction
Original classifier
SOH feature vector
Classification algorithms: Random Forest, SVM
Rectification
Total probability calculation
Weight vector w
Prediction vector p
t = <p, w>
Rectified by thresholds $t_1$ and $t_2$
Output class: Artcode or non-Artcode

**Fig. 3.** MR-augmented classification framework. The framework includes three stages: Follow-up generation, Prediction, and Rectification.


(a) Separation masks

(b) Occlusion masks

**Fig. 4.** Separation and occlusion masks.

second, train classification algorithms to predict the class of individual inputs. Xu et al. (2018) proposed two MRs, Separation and Occlusion, through examination of the differences in aggregated probability of image blocks being classified as *Artcode* between the Artcode and non-Artcode class. The two MRs were then used to enhance the classifier's performance based on conventional classification methods by adding a step before and after classification by this *base* classifier (referred to as the *original* classifier), resulting in an *MR-enhanced classifier*.

We have refined the MR-augmented classifier previously proposed in Xu et al. (2018) to present a new use of MRs in the verification of its correctness. As shown in Fig. 3, the MR-augmented classifier framework includes three stages: Follow-up generation, Prediction, and Rectification. Follow-up generation involves building inputs for the prediction stage using MR-defined image transformations. The second stage makes predictions about these inputs using commonly-used classification models. The third stage may adjust or rectify the results generated in the prediction stage. These three stages are described in detail in Section 3.1 to Section 3.3.

### 3.1. Follow-up generation

The core activity of the follow-up generation stage is to identify MRs and to construct the inputs based on the defined MRs. The identification of MRs in image classification is often done by examining the different image transformations, such as translation, rotation, and scaling. Based on the observation that the image blocks of Artcode images are more likely to be classified as *Artcode* than the blocks of non-Artcodes, two MRs, Separation and Occlusion, were proposed, using straightforward image operations: *uniform* and *non-uniform* separation. This stage accepts an entire image as input, and outputs image blocks generated from the operations defined by the two MRs.

#### 3.1.1. Separation MR

Separation involves splitting the input image uniformly into a number of sections, or *blocks*. For example, Fig. 4(a) shows separation masks to generate four uniform blocks by intersecting them with input images. This MR is based on the observation that the blocks of Artcode images would be predicted to be Artcode with a higher *likelihood* than the blocks of non-Artcode images. If the number of blocks is appropriately selected, this difference in the aggregated likelihood (probability) of all blocks may provide clues for classification. The Separation MR can be formulated as:

$$\sum_{i=1}^{n} \Pr(B^i_{s_a}) \geq \sum_{i=1}^{n} \Pr(B^i_{s_n}) \tag{1}$$

where $n$ is the number of image blocks; $\Pr(\cdot)$ is the probability for it to be classified as an Artcode by the original classifier; and $B^i_{s_a}$ and $B^i_{s_n}$ denote the $i$th block of the Artcode and non-Artcode images after separation, respectively.

#### 3.1.2. Occlusion MR

Occlusion is similar to separation, except that the image blocks are not split uniformly — overlapping among image blocks is permitted. As shown in Fig. 4(b), four occlusion masks are provided to intersect with the input image, so that the image blocks outlined by the white regions will be generated. Occluded Artcode images generally preserve the properties of the input Artcode images — half of an Artcode image usually has a higher likelihood of being classified as Artcode by the original classifier than a quarter of the image; this property may not be preserved for non-Artcode images: occluded non-Artcode images may have the equivalent likelihood as the entire non-Artcode images of being predicted as non-Artcode. Based on this observation, MR Occlusion can be formulated as:

$$\sum_{i=1}^{m} \Pr(B^i_{o_a}) \geq \sum_{i=1}^{m} \Pr(B^i_{o_n}) \tag{2}$$

$$\Pr(B^i_{o_a}) \geq \Pr(B^i_{s_a}) \tag{3}$$

where $m$ is the number of masks; $B^i_{o_a} = \cap(I_a, M_i)$ and $B^i_{o_n} = \cap(I_n, M_i)$ outputs the overlapping areas of Artcode and non-Artcode images, $I_a$ and $I_n$, and the $i$th mask $M_i$; and $B^i_{o_a}$ and

$B^i_{o_n}$ denote the $i$th block of the Artcode and non-Artcode images generated after occlusion, respectively.

The Separation and Occlusion MRs are both processed by comparing the aggregated likelihood of predicting the generated image blocks with the probability of predicting the entire input image. They are based on the observation that the topological structure of an Artcode image, as a global property, may be preserved, even after splitting. Uniform separation with (separation) and without (occlusion) overlapping enable the generated image blocks to cover the possible distribution of Artcodes in an image, especially considering their freeform geometric shapes. In addition, the masks with varying sizes can adapt to the Artcodes' scales. Therefore, they complement each other, and are combined together to obtain a better augmentation performance.

### 3.2. Prediction

In order to predict the class of an input image or block, a classification model that includes feature vector and classification algorithms (using random forests or support vector machines) needs to be built. The Artcode classification model is built using the Shape of Orientation Histograms (SOH) feature vector (Xu et al., 2017), which was specially designed for describing topological visual markers such as Artcodes. An SOH is constructed based on the *translational symmetry* and *smoothness* of the orientation histogram, which is a feature vector developed by McConnell (1986) for pattern analysis in both static and dynamic modes, and was adopted by Freeman and Adelson (1991) for recognising hand gestures.

Instead of describing the local geometry or structure, an SOH describes Artcodes by representing their topological structure. As previously reported (Xu et al., 2017), the orientation histogram of an Artcode displays horizontally translational symmetry, and is relatively smoother than that of a non-Artcode. The SOH is then constructed by quantifying these two aspects of the orientation histogram of the input images using similarity measurements, such as Procrustes distance (Moser, 1965) and $\chi^2$ distance (Greenacre, 2007). When all images are represented by their respective SOH vectors, classification algorithms using random forests or SVM are trained and used to predict the classes of the input images. The output of the prediction stage is a vector of labels of the input image blocks fed by the follow-up generation stage. This vector is referred to as the *prediction vector* $\vec{p}$.

### 3.3. Rectification

Unlike most deterministic software, classification is based on *statistics*, or is learned from past experience. Given an input, the output of a classifier is a *probabilistic* classification of belonging to a predefined class. In other words, before execution of the classifier, only the likelihood of the input being classified as a class or not is known beforehand. Therefore, in order to enable incorporation of the MRs described above, an augmented classifier integrating the MRs was designed based on *probability*, adding an adjustor (or rectifier) to the conventional classification pipeline (Xu et al., 2018).

As defined in Eqs. (1) to (3), the likelihood of image patches belonging to the two classes, generated in the follow-up generation stage, may be different. Therefore, a weight vector that contains different *weight* (i.e., likelihood) values is assigned to them. This vector, which has same dimensionality as the prediction vector $\vec{p}$, is referred to as the *weight vector* $\vec{w}$.

Given a prediction vector $\vec{p} = (p_{s_1}, \ldots, p_{s_n}, p_{o_1}, \ldots, p_{o_m})$, and a weight vector $\vec{w} = \left(w_{s_1}, \ldots, w_{s_n}, w_{o_1}, \ldots, w_{o_m}\right)$ – where $p_i$ is the predicted class of the $i$th image patch by the original classifier; $w_i$ is the weight assigned to the $i$th image patch (which is, in fact, the weight of the separation or occlusion mask); and $n$ and $m$ are the numbers of image patches generated by the two MRs – the inner product of $\vec{p}$ and $\vec{w}$ is the *aggregated likelihood* of belonging to the Artcode class ($\jmath$-value), which is defined as:

$$\vec{\rho} = \langle \vec{p}, \vec{w} \rangle = \left( \sum_{i=1}^{n} p_{s_i} \cdot w_{s_i} + \sum_{i=1}^{m} p_{o_i} \cdot w_{o_i} \right) \tag{4}$$

The aggregated likelihood is also known as the *total probability* (Xu et al., 2018; Xu, 2019). The augmented classifier predicts the label of the input by comparing the $\rho$-value with the given thresholds $t_1$ and $t_2$, using the following decision rules: if $\jmath < t_1$, then it is a non-Artcode; if $\jmath \geq t_2$, then it is an Artcode; otherwise, the input retains the original classifier's prediction result.

## 4. Experimental studies

This section presents the experimental study, including the evaluation dataset and the set-up of the experiment. The experimental results of verifying and enhancing the original classifier, and the performance comparison between the RF-based and SVM-based classifiers, are also described in this section.

### 4.1. Dataset

In order to study the Artcode classification problem, a dataset containing 47 Artcode and 116 non-Artcode images was used for experimental study. To the best of our knowledge, this is the first dataset available for studying Artcode classification. The non-Artcode images (including logos, drawings, advertisements, paintings, and graphics) were all created by humans, and were intentionally selected such that they would appear very similar to actual Artcode images (Xu et al., 2018). This means that the dataset is very challenging for Artcode classification. As shown in Figs. 5 and 6, Artcode examples look very similar to the non-Artcode images, which can make it very difficult to distinguish between the two classes through visual inspection alone. Because Artcodes are manually created by designers, the number of available Artcodes is currently small and slightly imbalanced, but work is ongoing to extend the dataset.[2] However, it is not possible to create hundreds of Artcode samples within a short time frame, much less increase the number to thousands or millions, like other common image classification tasks. Rather than devoting the very large effort necessary to expand the size of the dataset, we accepted this situation (of a small, imbalanced dataset), and adopted measures to address it, and mediate its impact: (1) We used classification methods that are effective on small datasets; (2) we adopted a group of carefully-considered performance evaluation metrics that are capable of evaluating classifiers used on imbalanced datasets; (3) we employed cross-validation techniques for experimental evaluation; and (4) we applied appropriate statistical methods to verify whether or not the improved performance was indeed attributable to the MR augmentation.

### 4.2. Cross-validation

Cross-validation is a commonly-used model validation technique for assessing how a learning model will generalise to a dataset (Kohavi, 1995; Devijver and Kittler, 1982; Seni and Elder, 2010). A major reason for using cross-validation, rather than using the conventional validation method that partitions the dataset into two sets (70% for training and 30% for testing), is that sufficient data may not be available for training and testing the

[2] https://www.artcodes.co.uk/creations/.

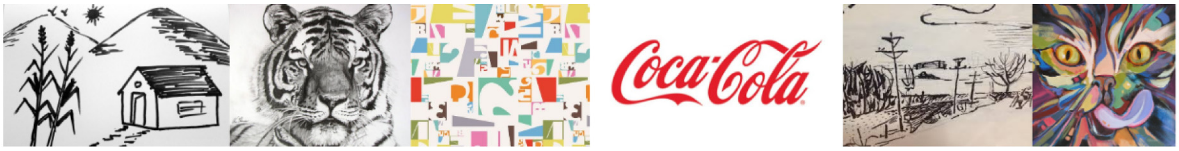


**Fig. 5.** Non-Artcode examples selected from the Artcode dataset.

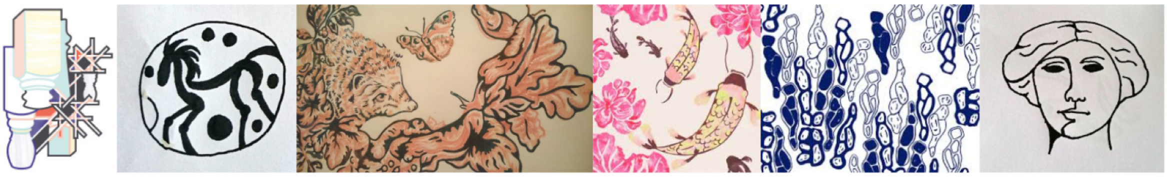


**Fig. 6.** Artcode examples selected from the Artcode dataset. Artcodes are visually "hidden" or even "invisible" markers. Similar to barcodes and QR codes, they can be scanned to trigger the digital information attached within. The code embedded in an Artcode is a string of numbers of *blobs* in each "hollow" *region*. For example, the code of the 6th image is "1-1-1-1-2".

model without compromising its generalisation and prediction capability.

Considering the limited number of samples in the Artcode dataset, a 5-fold cross-validation was used to ensure sufficient training and testing set sizes for performance evaluation. A $k$-fold cross-validation involves randomly partitioning a dataset into $k$ equally-sized subsets, keeping one subset as validation data for testing the trained model, and using the remaining $k-1$ subsets as training data. The process is then repeated $k$ times (the *folds*).

### 4.3. Study 1 — Verification

MT attempts to verify the software through examination of whether or not the identified MRs are violated: as explained in Section 2.1, violation of the Separation and/or Occlusion MR would indicate that the original classifier has not been correctly implemented.

Due to the *uncertainty* of a prediction by the original classifier, we explored its correctness by examining the weighted sum of probability of all image blocks of an input image being classified as *Artcode* – the aggregated likelihood $\rho$ – seeing if Artcodes and non-Artcodes had significant differences in the aggregated likelihood. Given input groups of $N$ Artcode and $M$ non-Artcode images, after the follow-up generation and prediction stages (Fig. 3), the two classes then have two sets of $\rho$-values calculated based on Eq. (4):

$$\rho_{G_a} = \left\{\rho_{a_i} \mid i = 1, \ldots, N\right\}, \quad \rho_{G_n} = \left\{\rho_{n_i} \mid i = 1, \ldots, M\right\} \tag{5}$$

where $\rho_{G_a}$ and $\rho_{G_n}$ denote the sets of aggregated likelihood of image samples of Artcode ($G_a$) and non-Artcode ($G_n$) category, respectively.

We then examined the implementation correctness by checking whether or not the relationship that $\rho_{G_a}$ and $\rho_{G_n}$ are significantly different was violated. Because of the probabilistic nature of the classifier, we used one-way analysis of variance (ANOVA) to assess the possible violation. ANOVA is a form of statistical hypothesis-testing that can be used to analyse whether or not there are statistically significant differences among the means of independent groups. We used ANOVA to examine if there was a statistically significant difference between the two groups $\rho_{G_a}$ and $\rho_{G_n}$ – overall, the $\rho_{G_a}$ may be *significantly* "greater" than $\rho_{G_n}$ from a statistical perspective – using separation and occlusion. If not, the classifier may be incorrectly implemented. When employing one-way ANOVA, it is assumed that the variances of different groups are equal and that the $\rho$-values are normally distributed. However, although the two groups were independently selected and members in groups $G_{n_i}$ were randomly selected, it was not certain that the *normality* and *equal variance* assumptions were satisfied in the experiment. Although one-way ANOVA is not very sensitive to deviations from normality, according to simulation results by McDonald (2009, pp. 157–164), we conducted further studies to consider situations of *non-normality* and *unequal variances*. In contrast to examining if the two assumptions were satisfied, we consolidated the experiment by introducing two more statistical test methods: t-test (for unequal variances) – which can be used to determine if the means of two groups $\rho_{G_n}$ and $\rho_{G_{n_i}}$ are significantly different when the variances are unequal; and Kruskal–Wallis test (Kruskal and Wallis, 1952) (also called one-way ANOVA on ranks, denoted ANOVA_ranks) – which is suitable for studying the difference between the means of two groups under non-normality situations. Hence, one-way ANOVA in conjunction with t-test and ANOVA_ranks can effectively evaluate the difference between the mean $\rho$-values of the two groups under the aforementioned situations. As the comparisons between $\rho_{G_a}$ and each $\rho_{G_n}$ using these three methods were conducted separately, rather than simultaneously, we also used Dunnett's test (Dunnett, 1955) as a *post hoc test* method. Dunnett's test is a *multiple comparison* procedure that enables one-to-many comparisons *simultaneously* to check if significant differences exist between the Artcode group $\rho_{G_a}$ and each of the non-Artcode groups $\rho_{G_{n_i}}$. The following sections present this verification examination, including detailing the experimental setting and results.

#### 4.3.1. Experimental setting

In order to examine the correctness of the classifier, we checked for violation of the MRs through examination of the variation of $\rho$-values between the two classes. Considering the different sizes of $G_a$ and $G_n$ ($N < M$) – $G_n$ is considerably larger than $G_a$ – $N$ elements were randomly selected from $G_n$ each time, with this process run $K$ times to generate $K$ non-Artcode groups $G_{n_i}, i = 1 \ldots K$. One-way ANOVA, t-test (for unequal variances), one-way ANOVA on ranks, and Dunnett's test were conducted to examine if there was a significant difference between $\rho_{G_a}$ and each $\rho_{G_{n_i}}$. To reduce variance, we randomly selected $K$ groups, $G_{n_i}, i = 1 \ldots K$, from the non-Artcode group $G_n$, in which each $G_{n_i}$ had the same size as the group $G_a$. We used the RF-based original classifier as the SUT for study, and a 5-*fold* cross-validation to obtain the prediction results of the image blocks generated by the follow-up generation stage. The weights of $w_{s_i}$ $(i = 1 \ldots n)$ and $w_{o_j}$ $(j = 1 \ldots m)$ were all assigned the same values, meaning that all image blocks generated based on separation or occlusion had the same weights — having the same likelihood to contain Artcodes. The weights between the images blocks for separation may be different from those for occlusion.

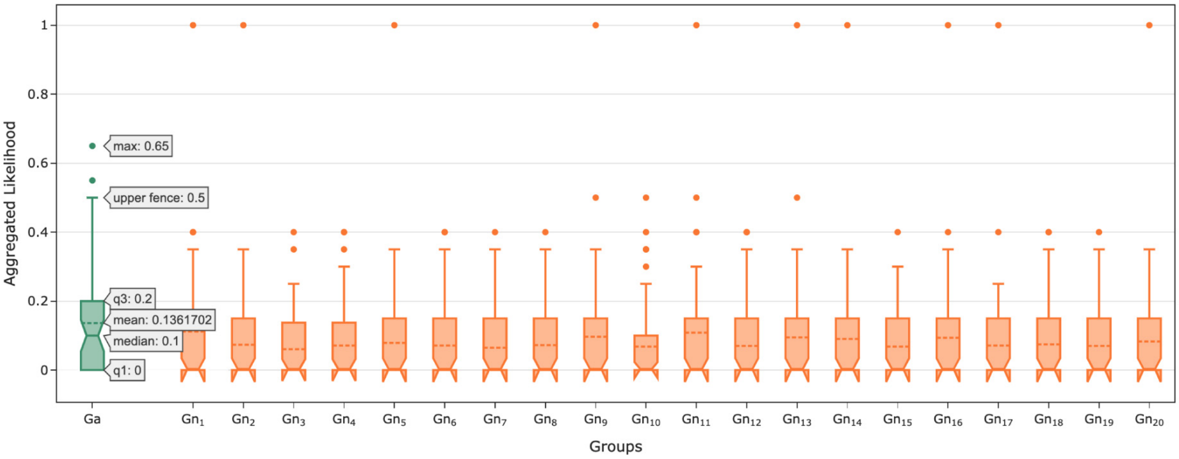


**Fig. 7.** Boxplot of the aggregated likelihood ($\rho$) of Artcode group ($G_a$) and non-Artcode groups ($G_{n_{1-20}}$). The dashed line in each box denotes the mean aggregate likelihood of the group, i.e., $\overline{\rho_G}$. The grey arrowed box annotations show the mean, maximum, minimum, median, first quartile (q1) and third quartile (q3) of the Artcode group.

### 4.3.2. *Results*

Fig. 7 presents a boxplot of the aggregated likelihoods of the group $G_a$ and $G_{n_{1-20}}$; and Table 1 shows the p-values for comparisons between $\rho_{G_a}$ and each $\rho_{G_{n_i}}$, according to the four tests. The average aggregated likelihoods (dashed line in Fig. 7) of all images in Artcode and non-Artcode categories were calculated using the following formula:

$$\overline{\rho_G} = \frac{1}{N}\sum_{i=1}^{N}\rho_i \tag{6}$$

where $\overline{\rho_G}$ is the mean aggregated likelihood of group $G$. The mean aggregated likelihood of all randomly generated non-Artcode groups $G_{n_i}$ is defined as:

$$\overline{\rho_{G_n}} = \frac{1}{K}\sum_{i=1}^{K}\overline{\rho_{G_{n_i}}} \tag{7}$$

The mean aggregated likelihood of all groups is defined as:

$$\overline{\rho_{G_a} + \rho_{G_n}} = \frac{1}{K+1}\left(\sum_{i=1}^{K}\overline{\rho_{G_{n_i}}} + \overline{\rho_{G_a}}\right) \tag{8}$$

where $K$ is the number of groups randomly selected from $G_n$; and $N$ and $M$ are the total number of Artcode and non-Artcode images in the Artcode dataset, respectively. We set $K$ to 20, which means that 20 groups were randomly selected for study. Both $n$ and $m$, the number of masks used in separation and occlusion, were set to 4.

As shown in Fig. 7, the $\rho$-value of the Artcode group is much less dispersed than that of the non-Artcode groups, showing less distance between the median and mean $\rho$-value. The mean aggregated likelihood data ($\overline{\rho_G}$) (denoted by dashed lines in the boxes) shows $\overline{\rho_{G_a}}$ (0.136170) to be greater than all the $\overline{\rho_{G_{n_i}}}$, $i = 1 \ldots 20$. This shows that, overall, the sum of probabilities of all image blocks of an Artcode image is greater than that of a non-Artcode image – indicating that the MRs have not been violated. Because of the uncertain nature of supervised classification, the aggregated likelihood of an individual Artcode image is *not* always greater than that of a non-Artcode image – the classifier may not predict inputs with 100% accuracy. However, the statistical analysis of variations between the groups $G_a$ and $G_n$ provides evidence for the difference of the mean $\rho$-values between Artcode and non-Artcode groups, indicating no violation of the MRs.

Table 1 presents the significance level (p-values) of the difference between $\rho_{G_a}$ and $\rho_{G_{n_i}}$ under ANOVA, t-test (for unequal variances), ANOVA_ranks, and Dunnett's test. Descriptive statistics – median, mean, minimum, maximum, and standard deviation (std) – for the p-values are also included. For ease of understanding, cells in the table are coloured to reflect the significance level: $p \leq 0.05$ are shown in dark gray; $0.05 < p \leq 0.10$ are in light gray; and $p > 0.10$ are in white. If the *null hypothesis* is defined as "an MR is violated", then small p-values (typically below 0.05) indicate strong evidence against the null hypothesis — small p-value indicate that neither of the two MRs have been violated. On the other hand, large p-values indicate weak evidence to reject the null hypothesis: there is no significant difference between the mean $\rho$-values of the $\rho_{G_a}$ and $\rho_{G_n}$ groups, under the chosen significance level, suggesting that one or both of the MRs may have been violated and, thus, the RF-based original classifier may have defects.

As shown in Table 1, the ANOVA p-values range from 0.000929 to 0.563244, with a median of 0.048375. Half of the $\rho_{G_n}$ groups show p-values that are considerably less than 0.05, indicating that these groups ($\rho_{G_{n_i}}, i = 1-3, 5, 8-9, 12,$ and $15-17$) are significantly different from $\rho_{G_a}$, under the significance level of 0.05 ($\alpha = 0.05$). If we increase the alpha value to 0.1, then two thirds of non-Artcode groups $\rho_{G_{n_i}}$ have means that are significantly different from the Artcode group $\rho_{G_a}$. This result provides evidence that the difference between the two groups is not due to sampling errors or by chance. The p-values of the remaining pairs are greater than 0.05, ranging from 0.059978 to 0.563244, indicating that there is no significant difference between the mean $\rho$-values of the two groups under $\alpha = 0.05$. This result can be explained by the diversity of the non-Artcode images in the Artcode dataset — some appear very similar to Artcode images, so-called "Artcode-like" images (Xu, 2019). Therefore, the significance level of the difference between $\overline{\rho_{G_a}}$ and $\overline{\rho_{G_n}}$ may decrease if $\rho_{G_n}$ includes many Artcode-like images. This will be discussed further in Section 5.

The mean ANOVA p-value is 0.094821, which is considerably larger than the median value of 0.048375. This indicates the skewness of the p-values: most p-values approach the minimal p-value, evidenced by the relatively higher standard deviation (0.137489). Although the mean p-value is relatively high (greater than the commonly-used significance level of 0.05), the low median p-value is evidence against the null hypothesis, reflecting the observed differences between $\overline{\rho_{G_a}}$ and most $\overline{\rho_{G_{n_i}}}$.

The one-way ANOVA results show that, even without assurance of equal variances and normality, $\rho_{G_a}$ is, to some extent, significantly different from $\rho_{G_{n_i}}$. Moreover, this significant difference was also observed under the assumptions of unequal

**Table 1**
Results of verification statistical analyses.

| $\rho_{G_n}$ | One-way ANOVA | t-test (for equal variances) | ANOVA_ranks (Kruskal–Wallis test) | Dunnett's test |
|---|---|---|---|---|
| 1 | 0.020599 | 0.020935 | 0.026708 | 0.137527 |
| 2 | 0.000929 | 0.001055 | 0.001055 | 0.006173 |
| 3 | 0.020941 | 0.021266 | 0.027382 | 0.005814 |
| 4 | 0.151530 | 0.151567 | 0.014303 | 0.022575 |
| 5 | 0.026392 | 0.026708 | 0.028257 | 0.016151 |
| 6 | 0.082988 | 0.082990 | 0.011025 | 0.025203 |
| 7 | 0.563244 | 0.563270 | 0.252606 | 0.011663 |
| 8 | 0.004583 | 0.004797 | 0.004104 | 0.023714 |
| 9 | 0.026213 | 0.026354 | 0.007432 | 0.024604 |
| 10 | 0.120316 | 0.120447 | 0.088321 | 0.004742 |
| 11 | 0.082720 | 0.083057 | 0.119343 | 0.198247 |
| 12 | 0.006003 | 0.006276 | 0.008328 | 0.013617 |
| 13 | 0.060633 | 0.060637 | 0.003846 | 0.040912 |
| 14 | 0.059978 | 0.060223 | 0.051375 | 0.029467 |
| 15 | 0.013469 | 0.013771 | 0.015104 | 0.034990 |
| 16 | 0.007758 | 0.008053 | 0.011156 | 0.089879 |
| 17 | 0.036771 | 0.037150 | 0.055180 | 0.005291 |
| 18 | 0.225829 | 0.226021 | 0.310495 | 0.042914 |
| 19 | 0.062216 | 0.062425 | 0.047335 | 0.018983 |
| 20 | 0.323313 | 0.323352 | 0.085770 | 0.029832 |
| median | 0.048375 | 0.048687 | 0.027045 | 0.024159 |
| mean | 0.094821 | 0.095018 | 0.058456 | 0.039115 |
| min | 0.000929 | 0.001055 | 0.001055 | 0.004742 |
| max | 0.563244 | 0.563270 | 0.310495 | 0.198247 |
| std | 0.137489 | 0.137422 | 0.083444 | 0.047734 |

variances and non-normality. The t-test (for unequal variances) has almost equivalent results to ANOVA (with only a negligible increase in p-values), thus supporting the same conclusion as ANOVA.

Table 1 also reports the results of the Kruskal–Wallis tests (ANOVA_ranks), which are suitable for non-normally distributed data. The ANOVA_ranks p-values are generally lower than those of ANOVA, ranging from 0.001055 to 0.310495, with a median of 0.027045 (which is less than the commonly-used $\alpha$-value of 0.05). 13 groups (1–6, 8–9, 12–13, 15–16, and 19) have p-values below 0.05. Compared with the ANOVA and t-test (for unequal variances) results, ANOVA_ranks has a considerably lower mean p-value (0.058456), which is only slightly greater than the $\alpha$-value of 0.05. The dispersion of p-values is also lower, with a smaller standard deviation of 0.083444. The ANOVA_ranks results confirm the significant differences between the means of $\rho_{G_a}$ and $\rho_{G_{n_i}}$ under the assumption of non-normality. This phenomenon could be explained by the ranked data type of the $\rho$-values: the $\rho$-values are not completely continuous, or normally distributed, but somehow show "ranks" in the proposed MR-augmented framework.

The p-values for one-way ANOVA, ANOVA_ranks, and t-test (for unequal variances) were calculated in separate comparisons. To alleviate the influence of this setting, and to consolidate the conclusion, we also conducted a multiple comparison test, Dunnett's test, to compare the Artcode group $\rho_{G_a}$ and the 20 non-Artcode groups $\rho_{G_{n_i}}$. Because the experiment studied the difference between $\rho_{G_a}$ and $\rho_{G_n}$, only the p-values for comparisons between $\rho_{G_a}$ and each $\rho_{G_{n_i}}$ are presented in Table 1. As can be seen from the table, Dunnett's test provides more evidence for significant differences between $\rho_{G_a}$ and $\rho_{G_n}$, with the p-values ranging from 0.004742 to 0.198247, and a median of 0.024159. 16 of the 20 groups were significantly different ($\alpha = 0.05$) from the Artcode group. In terms of mean and standard deviation, Dunnett's test had the lowest mean (0.039115) and standard deviation (0.047734) among all four tests. The results of the Dunnett's test thus confirm the significant difference between the Artcode group and non-Artcode groups.

Although none of the four test methods produced 20 p-values below 0.05, overall, the results in Table 1 show significant differences between the mean aggregated likelihoods of the Artcode and non-Artcode groups. Considering the uncertain nature of the predictor (the classifier) and the innate *variance* of random forests, the experimental results indicate no reason to consider the implementation faulty — the results indicate that neither MR has been violated. The next section will present the second study to evaluate the performance of the MR-augmented classifier, showing the enhanced performance of MR-augmented classifiers over non-augmented classifiers.

### 4.4. Study 2 — Enhancement

#### 4.4.1. Experimental setting

According to the framework in Fig. 3, we used Matlab to implement MR-augmented versions of classifiers that use random forests and support vector machines. The RF-based MR-augmented, SVM-based MR-augmented, RF-based non-MR-augmented (original) and SVM-based non-MR-augmented classifiers are denoted Aug-RF, Aug-SVM, Ori-RF and Ori-SVM, respectively. Cross-validation techniques were used to evaluate and compare the performance of these classifiers, with the Artcode dataset being used as the evaluation dataset.

Because random forests and SVM are used for the classification algorithms, the performance naturally has a certain level of variation in each execution — due to RF's random variable selection from the feature vector, and SVM's sub-optimisation because of the limited number of computational iterations. Multiple runs of cross-validation were therefore conducted to obtain the average performance. Because the dataset was imbalanced, with more non-Artcode than Artcode samples, we needed an appropriate group of measurements that could effectively deal with evaluation using imbalanced datasets to provide an informative view of the performance of the MR-augmented classifiers: Precision, recall, accuracy, the TNR (true negative rate), the $F_\beta$ measure, and the MCC (Matthews Correlation Coefficient) (Matthews, 1975) were all employed as evaluation metrics.

Precision is a measure of the correctness of those classified as Artcodes, whereas recall is a measure of completeness (how many of the true Artcodes were correctly classified). These two measures focus on positive examples and predictions, and their importance varies from one learning task to another. With Artcode classification, recall is more important than precision because recognising the presence of all Artcodes in the scene is a prerequisite to the follow-up decoding that triggers the digital information.

TNR measures how many non-Artcode samples are correctly classified. Accuracy, $F_\beta$, and MCC measure the overall performance of the classifier. Accuracy is the overall proportion of correct predictions, for both the positives (Artcodes) and negatives (non-Artcodes). However, accuracy is sensitive to size differences among classes, and, in our study, may have been influenced by the imbalanced class sizes. The $F_2$ measure is a special instance of the $F_\beta$ measure with $\beta = 2$, where $\beta$ is a value allocating $\beta$ times as much importance to recall as to precision. $F_2$ uses a weighted average of precision and recall to evaluate the classification effectiveness, giving twice ($\beta = 2$) as much importance to recall as to precision. In contrast to accuracy, the $F_2$ measure and MCC provide more insight into the performance of a classifier. However, compared with MCC, $F_2$ can be sensitive to data distribution. MCC is, in essence, a correlation coefficient between the observed and predicted classifications, incorporating true and false positives and negatives. It remains effective even if the dataset is imbalanced, and is generally regarded as one of the best measures for classification performance evaluation (Powers, 2011).

Two thresholds, $t_1$ and $t_2$, were studied in the experiment, as was their impact on the augmented classifiers. The given values in the weight vector $\vec{w}$ affect the selection of the values of $t_1$ and $t_2$. According to Eq. (3), the weights of image blocks generated by occlusion are greater than those generated by separation. In this experiment, four masks were used for both separation and occlusion ($n = m = 4$), resulting in both the prediction vector $\vec{p}$ and the weight vector $\vec{w}$ being 8-dimensional. Based on empirical examinations of assigning different values to $\vec{w}$, we assigned a value of 0.1 to both $w_{s_a}$ and $w_{s_n}$, and a value of 0.15 to both $w_{o_a}$ and $w_{o_n}$. In order to achieve quantisation and computational convenience of the value of aggregated likelihood $\rho$, the numbers 1 and 0 were used in the prediction vector $\vec{p}$ to represent the Artcode and non-Artcode classes, respectively.

### 4.4.2. Results

All performance metric values reported are the average values calculated from five executions of $k$-fold cross-validation. Two combinations of the two thresholds $t_1$ and $t_2$, in conjunction with different numbers of decision trees, were used to study the impact of the classifiers' tuning parameters. Because *nTrees* (the number of decision trees used in the RF-based classifiers) is not a tuning parameter of the SVM classifiers, for the sake of comparison, the SVM classifier values for each *nTrees* value are only the average of five runs of $k$-fold cross-validation. Higher values in Figs. 8 and 9 indicate better performance. Figs. 8 and 9 show a consistent performance across different values of *nTrees* for all six evaluation metrics: This means that the Aug-RF classifier (unbroken red) is not sensitive to changes in the value of *nTrees*, a characteristic inherited from the original RF classifier (dashed red).

*MR-augmented versus non-MR-augmented classifiers.* We studied the performance difference between the augmented (Aug-) and original (Ori-) classifiers, and also compared the performance of the classifiers based on random forests (-RF) with that of those based on support vector machines (-SVM).

Using various values of *nTrees* and fixed values of the thresholds $t_1$ and $t_2$, the MR-augmented classifiers (Aug-RF and Aug-SVM) outperformed the original classifiers (Ori-RF and Ori-SVM) in terms of precision, recall, accuracy, $F_2$, and MCC. They also outperformed the original classifiers in terms of recall, precision, and $F_2$ measure for threshold combinations of $t_1 = t_2 = 0.2$, and $t_1 = 0.15$, $t_2 = 0.3$, showing improved predictive performance in classification of the positive class (Artcodes). This improvement is important because Artcode classification requires higher accuracy when predicting Artcodes.

When predicting the negative class (non-Artcodes), as measured by TNR, the MR-augmented classifiers appear slightly influenced by different values of the thresholds ($t_1$ and $t_2$), which can be seen in the slight difference in TNR values for the original and augmented classifier in Figs. 8(d) and 9(d): for $t_1 = t_2 = 0.2$, the augmented classifier TNR values are similar to those for the original; but for $t_1 = 0.15$, $t_2 = 0.3$, they are less effective. This is different to the other evaluation metrics, which all show that the augmented classifiers outperform the original ones for both threshold combinations. A reason for this, partly as described in Section 3.3, is that when $t_1$ equals $t_2$, the augmented classifier does not directly use the prediction result of the original classifier. Another reason is the careful selection of threshold $t_1$: lower values of $t_1$ mean that the augmented classifier predicts the input image depending on the MRs only when they can adjust prediction with a relatively high confidence — otherwise, the augmented classifier uses the original prediction result. Thus, thresholds $t_1$ and $t_2$ can be used as tuning parameters for the performance of the MR-augmented classifier for both the positive and negative class.

Accuracy and MCC assess the overall performance of the classifier. As shown in Figs. 8(c) and 9(c), for both threshold combinations, the augmented classifiers have slightly better Accuracy than the original classifier, with an average increase of approximately 2%–3%. Although the MR-augmented classifiers show improved performance in the Artcode class, the small percentage of Artcodes in the dataset does not contribute strongly to the overall accuracy in evaluation, which is determined by both true positives and true negatives. In contrast, MCC is a more informative measure of overall performance, even when the dataset is imbalanced. As shown in Figs. 8(d) and 9(f), the augmented classifiers obtain about a 10%–20% increase over the original classifiers. This improvement is much more noticeable when comparing Aug-SVM with the Ori-SVM classifier, showing an overall improved performance of the MR-augmented classifier. However, the values of $F_2$ and MCC for all classifiers are relatively low. This is due to the imbalance of the dataset used in the evaluation, with a much greater number of negative examples than positive ones.

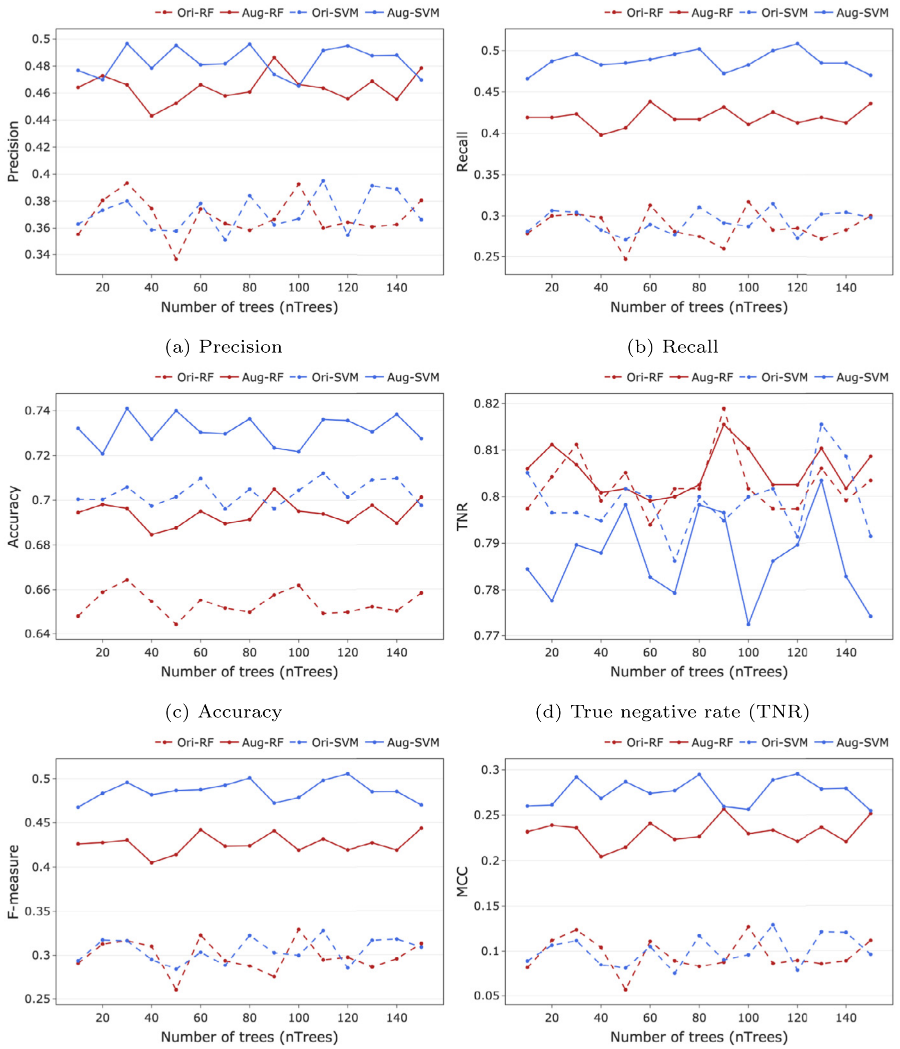
Both the original and MR-augmented classifiers achieve high true negatives (TN), approximately 0.82–0.85, as presented in Figs. 8(d) and 9(d). However, they also have very low true positives (TP), approximately 0.3–0.4, which can be observed from the low precision (Figs. 8(a) and 9(a)) and recall (Figs. 8(b) and 9(b)) results. If $TN = 0.85$ and $TP = 0.3$, then $FN = (1 - TN) = 0.15$ and $FP = (1 - TP) = 0.7$, and MCC can be calculated as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (9)$$

The MCC is a very low value, 0.1796. This illustrates how MCC is an effective measurement for evaluating the performance of a classifier on an imbalanced dataset.

As can be seen from Figs. 8 and 9, the precision and recall values of all classifiers are relatively low, and the TNR values are comparatively high. This is due to the imbalance in the Artcode dataset, which includes many more negatives. On the one hand, more weight is given to the non-Artcode class by feeding more information to the classification model in the training stage, resulting in a classifier with low recall evaluation (good non-Artcode classification, but poorer Artcode classification). On the other hand, the small percentage of Artcodes in the dataset results in the low precision evaluation of both classifiers. Conversely, the large proportion of non-Artcode images in the dataset (and the good non-Artcode prediction of the classifier) lead to relatively high TNR values, as shown in Figs. 8(d) and 9(d).

*RF-based versus SVM-based classifiers.* The SVM-based classifiers (blue lines) achieve better performance than the RF-based classifiers (red lines), as shown in Figs. 8 and 9, with an approximately 5%–10% increase in terms of almost all performance evaluation measurements (not for TNR). The tradeoff between the precision and TNR of the SVM-based classifiers can be adjusted by the misclassification matrix (Cortes and Vapnik, 1995) employed in



(a) Precision

(b) Recall

(c) Accuracy

(d) True negative rate (TNR)

(e) $F_2$ measure

(f) Matthews correlation coefficient (MCC)

**Fig. 8.** Performance comparison between RF and SVM-based classifiers with different values of *nTrees* and $t_1, t_2 = 0.2$.

SVM. Considering the greater importance of recall than precision in this application, this experiment assigned higher values to the cost of classifying an Artcode as a non-Artcode, resulting in a classifier that enables better Artcode prediction.

The better performance of the SVM-based classifiers is also evidenced by the higher values of the Aug-SVM classifier than the Aug-RF classifier. However, when the classifiers use the same classification method (SVM or RF), the MR-augmented version outperformed the original (non-augmented) version of the corresponding classifier. This indicates that the introduction of MRs into supervised classification models actually improves the performance of the original classifiers, regardless of whether SVM or RF is used.

Overall, the Aug-SVM classifier obtained the best performance, especially when considering that SVM runs much faster than the random forests classifier. The MR-augmented classifiers outperformed the original classifiers in terms of all the evaluation measures. This improved performance is sensitive to the values of the thresholds $t_1$ and $t_2$, but not to the value of *nTrees*, or the choice of classification method. As discussed in Section 4.4.2, thresholds $t_1$ and $t_2$ influence the performance of the augmented classifier, with different combinations determining the impact the MRs have on adjusting the original classification. Careful selection of the values of the tuning parameters – the thresholds $t_1$ and $t_2$ – is therefore vital to fine-tune the results of the original classifier and obtain the enhanced performance.

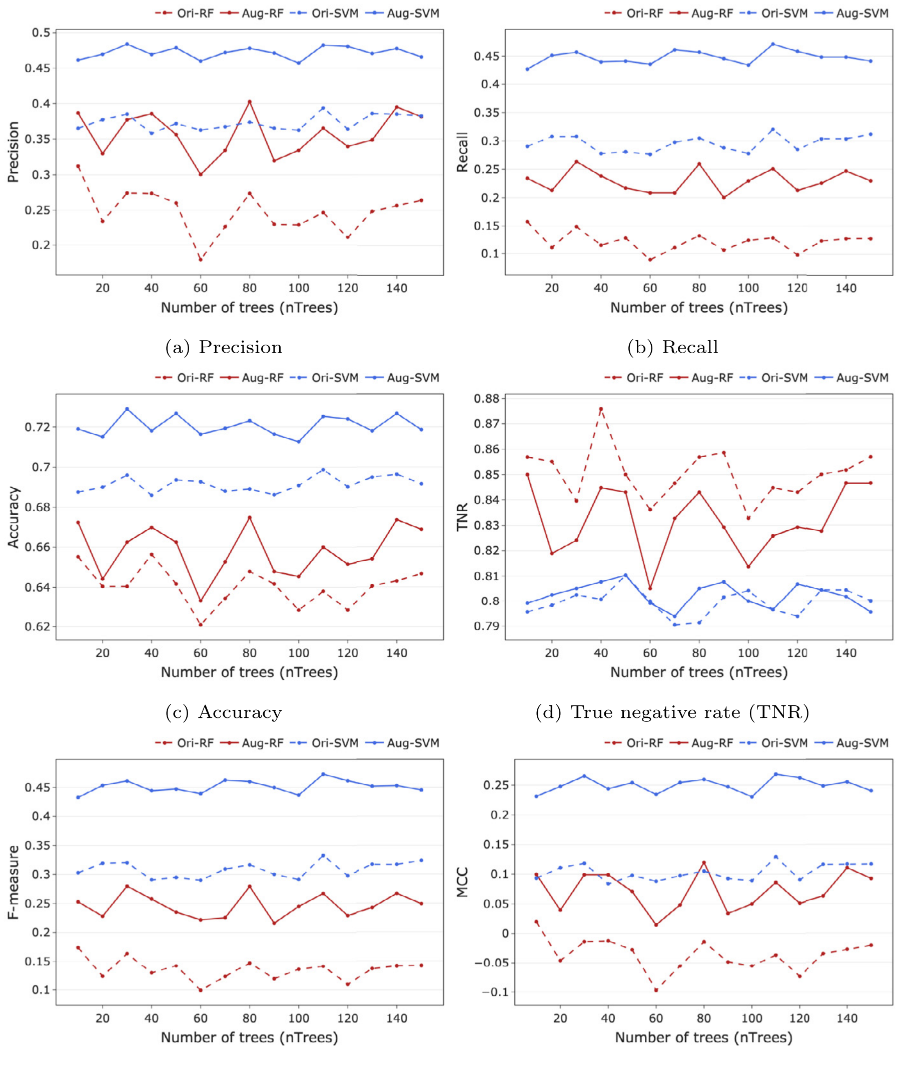
**Table 2**
Results of rectification analysis.

| Class | Amount | Aug-RF rectifications | | Aug-SVM rectifications | |
|---|---|---|---|---|---|
| | | Correct | Incorrect | Correct | Incorrect |
| Artcode | 47 | 13.3 (28.3%) | 1.9 (4.04%) | 13.8 (29.36%) | 4 (8.51%) |
| non-Artcode | 116 | 7.3 (6.3%) | 15.6 (13.45%) | 6.4 (5.52%) | 9.2 (7.93%) |

## 5. Analysis and discussion

### 5.1. Analysis of the rectification stage

In order to reveal how the fine-tuning (rectification layer) stage operates, and how the improved performance is achieved, we performed ten rounds of cross-validation runs using both the RF-based and SVM-based MR-augmented classifiers on all samples in the Artcode dataset. Table 2 shows the average correct and incorrect rectifications by the MR-augmented classifiers over these ten executions of 5-fold cross-validation. Fig. 10 shows $\rho$-values of all Artcodes ($\triangle$) and non-Artcodes ($\circ$) of one execution of cross-validation, where correct and incorrect rectifications are



(a) Precision

(b) Recall

(c) Accuracy

(d) True negative rate (TNR)

(e) $F_2$ measure

(f) Matthews correlation coefficient (MCC)

**Fig. 9.** Performance comparison between the RF- and SVM-based classifiers with different *nTrees* values and $t_1 = 0.15, t_2 = 0.3$.

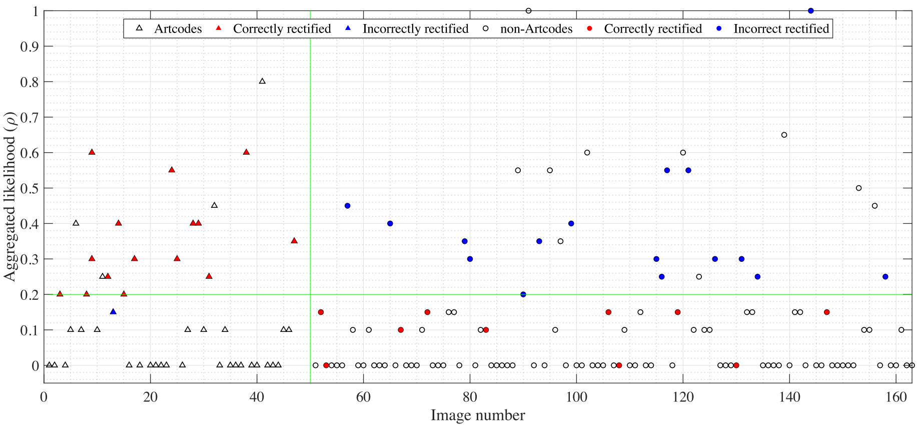


**Fig. 10.** Illustration of rectification distribution of the RF-based MR-augmented classifier. The graph is generated from one round of cross-validation of the RF-based MR-augmented classifier with *nTrees* = 30, and $t_1 = t_2 = 0.2$. This graph is split into left and right areas separated by a green vertical line, where the left and right area are an illustration of the aggregated likelihood ($\rho$-value) of Artcode ($\circ$) and non-Artcode ($\triangle$) images. The horizontal green line is the predefined thresholds ($t_1$ and $t_2$): It separates the graph into upper and lower zones. The samples in the upper zone ($\geq t_2$) are rectified as *Artcodes*, whereas the samples in the lower zone ($\leq t_1$) are labelled as *non-Artcodes* in the rectification stage of the MR-augmented classifier. Therefore, the two tuning parameters, thresholds $t_1$ and $t_2$, of the MR-augmented classifier control whether or not to rectify more "Artcode-like" samples. Correctly and incorrectly rectified predictions are highlighted in red and blue, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

highlighted in red and blue, respectively. As illustrated in Fig. 10 and Table 2, the two MR-augmented classifiers correctly rectified an average of 28.3% and 29.36% of the Artcode predictions, but incorrectly adjusted an average of 4.04% and 8.51% of the Artcodes to non-Artcodes. This higher correct rectification percentage contributed to the higher true positive rate — a key factor in the evaluation of a classifier in terms of recall and precision. However, the classifiers performed slightly worse on the non-Artcode class: the RF-based MR-augmented classifier had an average of 6.63% correct and 13.45% incorrect rectifications, and the SVM-based MR-augmented classifier obtained an average of 5.52% correct and 7.93% incorrect rectifications. This explains why the MR-augmented classifiers have a relatively lower true negative rate (TNR), as shown in Figs. 8(d) and 9(d), but higher precision (Figs. 8(a) and 9(a)) and recall (Figs. 8(b) and 9(b)). Overall, the average correct rectification percentage is 1.91% ( $\frac{13.3-1.9+7.3-15.6}{47+116} =$ 1.91%) for the Aug-RF classifier and 4.29% ( $\frac{13.8-8.51+6.4-9.2}{47+116} =$ 4.29%) for the Aug-SVM classifier, indicating that 1.91% and 4.29% of incorrect predictions by the RF-based and SVM-based original classifier were corrected by their respective MR-augmented classifiers. This explains how the improved performance of the MR-augmented classifiers was obtained: the rectification stage can rectify misclassifications (mainly false negatives) made by the original classifiers, albeit at the expense of comparatively fewer incorrect rectifications of true negatives.

The superior rectification performance of the MR-augmented classifiers on the Artcode examples shows that Artcode blocks are more likely to preserve the topological structure than non-Artcode blocks. Therefore, although the two MRs may violate the properties of Artcode images, the aggregated predictions of image blocks of Artcodes are more informative than the predictions of the entire image. This property may not be preserved for non-Artcodes, which have no predefined topological characteristics. The MR-augmented classifier adjusts prediction results in the rectification stage only if the new evidence collected is strong enough to accept, which is determined by comparing the aggregated likelihood of the predictions of image blocks with the given thresholds $t_1$ and $t_2$. Further discussion about how the MR-augmented classifier works is presented in the next section.

### 5.2. Discussion

As explained in Section 4.4.2, the MR-augmented classifiers obtained better recall and precision results than the original classifiers (with approximately 10%–15% improvement). Recall and precision focus on the positive class (Artcode), with higher values indicating more confident and complete predictions of Artcodes, while some Artcode misclassifications by the original classifier were corrected by the MR-augmented classifier in the rectification stage. The decision as to whether or not to rectify was based on the $\rho$-value, as given in Eq. (4), a measure of aggregated likelihood that an input image belongs to the Artcode class ( Section 3.3).

As described in Section 3.1, the Separation and Occlusion MRs are based on the assumption that Artcode image blocks are more likely to be classified as *Artcode* than *non-Artcode*. The effectiveness of the two MRs was investigated by examining the prediction and rectification of image blocks for those images adjusted by the MR-augmented classifier (the red and blue points in Fig. 10). The non-Artcodes that were incorrectly adjusted were the images that were very similar in topology to Artcodes (containing a number of connected regions), and had repeated geometrical structures, such as the 2nd and 4th images in Fig. 5. Repeated structures enabled the separate image blocks to inherit more topological structure from the original image, making their internal structures similar to those of Artcodes. Occlusion and separation sometimes strengthened their topological structure, because occlusion and separation may remove auxiliary structures such as background imagery. Accordingly, the MR-augmented classifiers are sensitive to this kind of *Artcode-like* images (such as the 4th image in Fig. 5) – images that are topologically very similar to Artcodes – which may result in incorrect rectifications.

Likewise, if separation and occlusion completely break the topological structure, Artcodes would be incorrectly rectified as non-Artcode by the MR-augmented classifiers. Fortunately, Artcodes have a topological structure that includes a number of connected regions, and often include several repeated structures with the same topology (but different geometry). These two properties enable Artcode image blocks to very likely retain the original topology, even after separation and occlusion. An example is presented in Fig. 11 for illustration: The image (the 5th in Fig. 6) is split into eight blocks by intersecting with the eight separation and occlusion masks shown in Fig. 4 — the left four image blocks in Fig. 11 are from separation, and the right four are from occlusion. Almost all of these blocks retain a complete topological structure: they remain relatively complete Artcodes. Therefore, the MR-augmented classifier, based on the aggregated probability ($\rho$-value) of image blocks belonging to the *Artcode* class, can accumulate more information about this Artcode image than the original classifier, thereby achieving better overall predictions.

The two MRs are based on fundamental image processing operations, with the underlying rationale being whether or not the image blocks are able to retain the original structure's properties after transformations. Artcodes, as topological markers enabling redundancy, naturally possess this property. The conventional use of MRs in metamorphic testing draws on intrinsic properties of the SUT. Likewise, the MRs used in Artcode classification also make use of intrinsic characteristics of Artcodes and non-Artcodes. Because domain characteristics may differ from task to task, and the repeated structures used in our two identified MRs may not exist in some contexts, it is likely that these MRs may not be directly applicable in some other image classification tasks. Nevertheless, this study has shown that MRs do have the potential to be used in image classification tasks (or even more general machine learning tasks), especially for those tasks with distinctive structural properties among different categories of learning data.

## 6. Conclusion

This paper has reported on an examination of two previously identified MRs to enhance image classification, using them not only to improve performance, but also to explore verification of the classifier. Considering the uncertainty of classification algorithms, the verification exploration involved four statistical tests: one-way ANOVA, t-test (for unequal variances), Kruskal–Wallis test, and Dunnett's test. An effective and efficient MR-augmented classifier that uses SVM as the classification method, Aug-SVM, was introduced, and was compared with the Aug-RF classifier. The paper also examined the MR-augmented classification framework (Xu et al., 2018), and presented a method that could be applied to related image classification problems for verification and enhancement.

Our experimental studies showed the applicability of ANOVA in conjunction with t-test (for unequal variances), ANOVA_ranks, and Dunnett's test to explore verification of the classifier based on the two MRs. The improved performance was not affected by the chosen classification method, demonstrating the potential to apply MT theories and techniques to general machine learning applications. Among the four classifiers in this paper (Ori-RF,

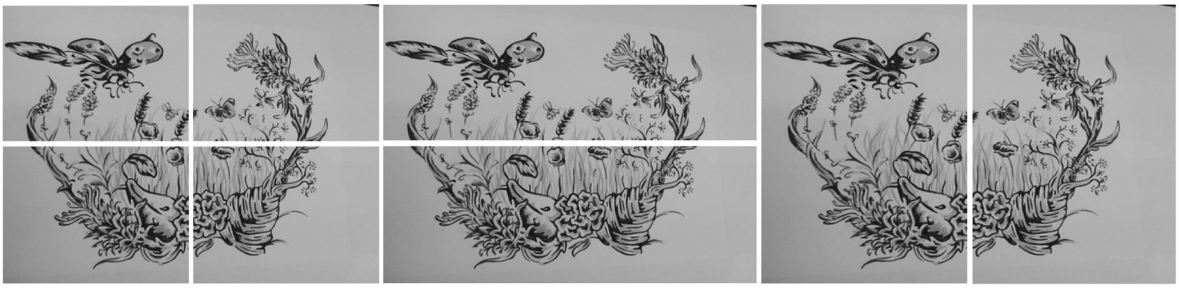


**Fig. 11.** Image blocks generated according to separation and occlusion.

Aug-RF, Ori-SVM, and Aug-SVM), Aug-SVM obtained the best performance in terms of both the evaluation metrics, and the computational efficiency. The experimental results also showed the essential role of the two thresholds, $t_1$ and $t_2$, for tuning the MR-augmented classifier performance. In addition, a theoretical analysis and discussion about how the enhanced performance was achieved by the MR-augmented classifiers was presented.

Our future work will include further examination of other parameters, including the number of masks ($n$ and $m$) for the separation and occlusion, the values in the weight vector $\vec{w}$, and the values of thresholds $t_1$ and $t_2$. A potential approach for choosing suitable values of $t_1$ and $t_2$ will be to examine the relationship between the thresholds and the centroids of $\rho_{G_a}$ and $\rho_{G_n}$. Because the work presented here has only examined two straightforward image transformations for the MRs, exploring other possible MRs that draw from other transformations for general image classification tasks, will also form part of our future work.

Although the two MRs employed in this work were straightforward, the results are promising, and clearly demonstrate the feasibility of MRs being used to augment classifiers. In order to fully investigate this new research area, more theoretical and practical work needs to be conducted, including exploration of connections between MRs and data augmentation, and case studies to examine the application of MRs to other well-studied image classification tasks (such as face and object detection) and even more broad machine learning problems. The concept of verifying machine learning software (the classifier in this paper) using MRs is still in an early stage of development, and more effort is also needed in the future. The proposed verification exploration based on ANOVA, t-test (for unequal variances), ANOVA_ranks, and Dunnett's test, attempts to use statistical analyses to test *probabilistic* algorithms such as classification models: Further work is necessary to extend this approach to verification, and fully explore its applicability.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China, under grant no. 61872167, by the Australian Research Council's Discovery Projects funding scheme (Project ID: DP210102447), and by a Western River entrepreneurship grant.

## References

Barr, Earl T., Harman, Mark, McMminn, Phil, Shahbaz, Muzammil, Yoo, Shin, 2015. The oracle problem in software testing: A survey. IEEE Trans. Softw. Eng. 41 (5), 507–525.

Benford, Steve, Hazzard, Adrian, Chamberlain, Alan, Glover, Kevin, Greenhalgh, Chris, Xu, Liming, Hoare, Michaela, Darzentas, Dimitrios, 2016. Accountable artefacts: the case of the Carolan guitar. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'16). ACM, San Jose, CA, USA, pp. 1163–1175.

Benford, Steve, Hazzard, Adrain, Chamberlain, Alan, Xu, Liming, 2015a. Augmenting a guitar with its digital footprint. In: Proceedings of International Conference on New Interfaces for Musical Expression (NIME'15), Louisiana, USA, pp. 303–306.

Benford, Steve, Hazzard, Adrian, Xu, Liming, 2015b. The Carolan guitar: a thing that tells its own life story. Interactions 22 (3), 64–66.

Benford, Steve, Koleva, Boriana, Preston, William Westwood, Angus, Alice, Thorn, Emily-Clare, Glover, Kevin, 2018. Customizing hybrid products. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 1–12.

Bergamasco, Filippo, Albarelli, Andrea, Rodola, Emanuele, Torsello, Andrea, 2011. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11). IEEE, Providence, RI, USA, pp. 113–120.

Chen, Tsong Yueh, Cheung, Shing Chi, Yiu, Shiu Ming, 1998. Metamorphic testing: a new approach for generating next test cases. Technical Report HKUST-CS98-01, Department of Computer Science, the Hong Kong University of Science and Technology.

Chen, Tsong Yueh, Feng, Jianqiang, Tse, T.H., 2002. Metamorphic testing of programs on partial differential equations: a case study. In: Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC'02). IEEE, Oxford, UK, pp. 327–333.

Chen, Tsong Yueh, Kuo, Fei-Ching, Liu, Huai, Poon, Pak-Lok, Towey, Dave, Tse, T.H., Zhou, Zhi Quan, 2018. Metamorphic testing: A review of challenges and opportunities. ACM Comput. Surv. 51 (1), 1–27.

Chen, Tsong Yueh, Kuo, Fei Ching, Ma, Wenjuan, Susilo, Willy, Towey, Dave, Voas, Jeffrey, Zhou, Zhi Quan, 2016. Metamorphic testing for cybersecurity. Computer 49, 48–55.

Chen, Tsong Yueh, Kuo, Fei-Ching, Towey, Dave, Zhou, Zhi Quan, 2015. A revisit of three studies related to random testing. Sci. China Inf. Sci. 58 (5), 1–9.

Chen, Tsong Yueh, Tse, T.H., Zhou, Zhi Quan, 2003. Fault-based testing without the need of oracles. Inf. Softw. Technol. 45 (1), 1–9.

Cortes, Corinna, Vapnik, Vladimir, 1995. Support vector networks. Mach. Learn. 20 (3), 273–297.

Costanza, Enrico, Huang, Jeffrey, 2009. Designable visual markers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09). ACM, pp. 1879–1888.

Costanza, Enrico, Robinson, John, 2003. A region adjacency tree approach to the detection and design of fiducials. In: Proceedings of Video Vision and Graphics Conference, pp. 63–69.

Devijver, Pierre A., Kittler, Josef, 1982. Pattern Recognition: A Statistical Approach. Prentice Hall, London, UK.

Dijkstra, Edsger W., 1972. Chapter I: Notes on structured programming. In: Structured Programming. Academic Press Ltd., UK, pp. 1–82.

Ding, Junhua, Zhang, Dongmei, Hu, Xinhua, 2016. An application of metamorphic testing for testing scientific software. In: Proceedings of the 1st International Workshop on Metamorphic Testing (MET'16). ACM, Austin, TX, USA, pp. 37–43.

Donaldson, Alastair F., Evrard, Hugues, Lascu, Andrei, Thomson, Paul, 2017. Automated testing of graphics shader compilers. Proceedings of the ACM on Programming Languages 1 (OOPSLA), 1–29.

Dunnett, Charles W., 1955. A multiple comparison procedure for comparing several treatments with a control. J. Amer. Statist. Assoc. 50 (272), 1096–1121.

Fiala, Mark, 2005. Artag, a fiducial marker system using digital techniques. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2. IEEE, San Diego, CA, USA, pp. 590–596.

Freeman, William T., Adelson, Edward H., 1991. The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. 13 (9), 891–906.

Greenacre, Michael, 2007. Correspondence Analysis in Practice. Chapman and Hall/CRC.

He, Pinjia, Meister, Clara, Su, Zhendong, 2019. Structure-invariant testing for machine translation. ArXiv, arXiv:abs/1907.08710.

Hughes, John, 2020. How to specify it!. In: Bowman, William J., Garcia, Ronald (Eds.), Trends in Functional Programming. Springer International Publishing, Cham, pp. 58–83.

International Organization for Standardization, 2015. Information technology – automatic identification and data capture techniques – QR code bar code symbology specification. ISO/IEC 18004:2015, ISO, https://www.iso.org/standard/62021.html.

Kanewala, Upulee, Bieman, James M., 2013. Using machine learning techniques to detect metamorphic relations for programs without test oracles. In: Proceedings of the 24th International Symposium on Software Reliability Engineering (ISSRE'13). IEEE, Pasadena, CA, USA, pp. 1–10.

Kohavi, Ron, 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95). 2, Morgan Kaufmann Publishers, Montreal, Quebec, Canada, pp. 1137–1143.

Koleva, Boriana, Spence, Jocelyn, Benford, Steve, Kwon, Hyosun, Schnädelbach, Holger, Thorn, Emily, Preston, William, Hazzard, Adrian, Greenhalgh, Chris, Adams, Matt, Farr, Ju Row, Tandavanitj, Nick, Angus, Alice, Lane, Giles, 2020. Designing hybrid gifts. ACM Trans. Comput.-Hum. Interact. 27 (5).

Kruskal, William H., Wallis, W. Allen, 1952. Use of ranks in one-criterion variance analysis. J. Amer. Statist. Assoc. 47 (260), 583–621.

Le, Vu, Afshari, Mehrdad, Su, Zhendong, 2014. Compiler validation via equivalence modulo inputs. In: Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'14). ACM, New York, NY, USA, pp. 216–226.

Lindvall, Mikael, Ganesan, Dharmalingam, Árdal, Ragnar, Wiegand, Robert E., 2015. Metamorphic model-based testing applied on NASA DAT: An experience report. In: Proceedings of the 37th International Conference on Software Engineering (ICSE'15). 2, IEEE, Florence, Italy, pp. 129–138.

Liu, Huai, Kuo, Fei-Ching, Towey, Dave, Chen, Tsong Yueh, 2014. How effectively does metamorphic testing alleviate the oracle problem? IEEE Trans. Softw. Eng. 40 (1), 4–22.

Ma, Pingchuan, Wang, Shuai, Liu, Jin, 2020. Metamorphic testing and certified mitigation of fairness violations in NLP models. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. International Joint Conferences on Artificial Intelligence Organization, pp. 458–465.

Mai, Phu X., Pastore, Fabrizio, Goknil, Arda, Briand, Lionel, 2020. Metamorphic security testing for web systems. In: 2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST). IEEE, Porto, Portugal, pp. 186–197.

Matthews, Brian, 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochim. Biophys. Acta (BBA)-Protein Structure 405 (2), 442–451.

Mayer, Johannes, Guderlei, Ralph, 2006. On random testing of image processing applications. In: Proceedings of the 6th International Conference on Quality Software (QSIC'06). IEEE, Beijing, China, pp. 85–92.

McConnell, Robert K., 1986. Method of and apparatus for pattern recognition. Google Patents, US Patent 4, 567, 610.

McDonald, John H., 2009. Handbook of Biological Statistics, vol. 2. Sparky House Publishing Baltimore, Maryland.

McNutt, Andrew, Kindlmann, Gordon, Correll, Michael, 2020. Surfacing visualization mirages. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI'20). Association for Computing Machinery, New York, NY, USA, pp. 1–16.

Meese, Rupert, Ali, Shakir, Thorne, Emily-Clare, Benford, Steve, Quinn, Anthony, Mortier, Richard, Koleva, Boriana, Pridmore, Tony, Baurley, Sharon L, 2013. From codes to patterns: designing interactive decoration for tableware. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'13). ACM, Paris, France, pp. 931–940.

Moser, Jürgen, 1965. On the volume elements on a manifold. Trans. Amer. Math. Soc. 120 (2), 286–294.

Murphy, Christian, Kaiser, Gail, Hu, Lifeng, Wu, Leon, 2008. Properties of machine learning applications for use in metamorphic testing. In: Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'08). Redwood City, CA, USA, pp. 867–872.

Ng, Kher Hui, Shaikh, Shazia Paras, 2016. Design of a mobile garden guide based on artcodes. In: Proceedings of 2016 International Conference on User Science and Engineering (I-USEr'16). IEEE, Puchong, Malaysia, pp. 23–28.

Powers, David, 2011. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. J. Mach. Learn. Technol. 2 (1), 37–63.

Preston, William, Benford, Steve, Thorn, Emily-Clare, Koleva, Boriana, Rennick-Egglestone, Stefan, Mortier, Richard, Quinn, Anthony, Stell, John, Worboys, Michael, 2017. Enabling hand-crafted visual markers at scale. In: Proceedings of the 2017 Conference on Designing Interactive Systems (DIS'17). Association for Computing Machinery, New York, NY, USA, pp. 1227–1237.

Ribeiro, Marco Tulio, Wu, Tongshuang, Guestrin, Carlos, Singh, Sameer, 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20), pp. 4902–4912.

Segura, Sergio, Fraser, Gordon, Sanchez, Ana, Ruiz-Cortés, Antonio, 2016. A survey on metamorphic testing. IEEE Trans. Softw. Eng. 42, 805–824.

Seni, Giovanni, Elder, John, 2010. Ensemble methods in data mining: improving accuracy through combining predictions. Synth. Lect. Data Min. Knowl. Discov. 2 (1), 1–126.

Thorn, Emily-Clare, Rennick-Egglestone, Stefan, Koleva, Boriana, Preston, William, Benford, Steve, Quinn, Anthony, Mortier, Richard, 2016. Exploring large-scale interactive public illustrations. In: Proceedings of the 2016 ACM Conference on Designing Interactive Systems (DIS'16). ACM, Brisbane, Australia, pp. 17–27.

Woodland, Norman J., Bernard, Silver, 1952. Classifying apparatus and method. Google Patents, US Patent 2, 612, 994.

Xie, Xiaoyuan, Ho, Joshua, Murphy, Christian, Kaiser, Gail, Xu, Baowen, Chen, Tsong Yueh, 2009. Application of metamorphic testing to supervised classifiers. In: Proceedings of the 9th International Conference on Quality Software (QSIC'09). IEEE, Jeju, Korea, pp. 135–144.

Xie, Xiaoyuan, Ho, Joshua W.K., Murphy, Christian, Kaiser, Gail, Xu, Baowen, Chen, Tsong Yueh, 2011. Testing and validating machine learning classifiers by metamorphic testing. J. Syst. Softw. 84 (4), 544–558.

Xu, Liming, 2019. Artcode detection in images (Ph.D. thesis). School of Computer Science, University of Nottingham, http://eprints.nottingham.ac.uk/59474/.

Xu, Liming, French, Andrew P., Towey, Dave, Benford, Steve, 2017. Recognizing the presence of hidden visual markers in digital images. In: Proceedings of the on Thematic Workshops of ACM Multimedia 2017. In: Thematic Workshops '17, ACM, Mountain View, CA, USA, pp. 210–218.

Xu, Liming, Towey, Dave, French, Andrew P, Benford, Steve, Zhou, Zhi Quan, Chen, Tsong Yueh, 2018. Enhancing supervised classifications with metamorphic relations. In: 2018 IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET'18). IEEE, Gothenburg, Sweden, pp. 46–53.

Zhang, Mengshi, Zhang, Yuqun, Zhang, Lingming, Liu, Cong, Khurshid, Sarfraz, 2018. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering (ASE'18). ACM, New York, NY, USA, pp. 132–142.

Zhou, Zhi Quan, Sun, Liqun, 2018. Metamorphic testing for machine translations: MT4MT. In: Proceedings of the 25th Australasian Software Engineering Conference (ASWEC'18). IEEE, Adelaide, Australia, pp. 96–100.

Zhou, Zhi Quan, Sun, Liqun, 2019. Metamorphic testing of driverless cars. Commun. ACM 62 (3), 61–67.

Zhou, Zhi Quan, Xiang, Shaowen, Chen, Tsong Yueh, 2016. Metamorphic testing for software quality assessment: A study of search engines. IEEE Trans. Softw. Eng. 42 (3), 264–284.