The Journal of Systems & Software 181 (2021) 111040

Contents lists available at ScienceDirect

The Journal of Systems & Software

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

Stability evaluation for text localization systems via metamorphic testing[☆]

Rongjie Yan [a,c,*], Siqi Wang [a,c], Yixuan Yan [b], Hongyu Gao [b], Jun Yan [a,c]

[a] State Key Laboratory of Computer Science, ISCAS, China
[b] College of Computer, Beijing University of Technology, China
[c] University of Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 30 November 2020
Received in revised form 2 July 2021
Accepted 6 July 2021
Available online 13 July 2021

Keywords:
Learning techniques
Text localization
Stability
Metamorphic testing

ABSTRACT

The success of learning techniques in solving a variety of hard AI problems promotes the flourish of recognition-based applications. Many state-of-the-art text localization systems, which can detect and report the positions of text segments in an image, are mainly implemented with learning-based techniques. Data-driven learning raises a series of questions on how to verify, validate and evaluate such learning-based systems. In this paper, we propose a methodology to automatically evaluate the stability of text localization systems via metamorphic relations, where a stable system should output consistent results for similar inputs with the same text segments. We introduce six metamorphic relations that should be preserved in a stable text localization system and define the corresponding metrics for stability evaluation. With the defined metamorphic relations, we apply metamorphic testing techniques to compare the inputs and outputs to evaluate system stability, and further diagnose the causes of inconsistency. The extensive experimentation on both academic and commercial text localization systems demonstrates the effectiveness of our method on stability evaluation for such systems.

1. Introduction

Nowadays, learning-based systems are omnipresent for the great success of deep learning techniques in solving a variety of hard AI problems. The performance of such techniques depends on various factors, such as the choice of training data, the structure of the system, and the process of training. Text localization and recognition, which can extract the location and the contents of a text segment from an image, is one of the popular learning-based applications (Lin et al., 0000). Such systems are sensitive to the quality of the inputs. The evaluation with traditional metrics on performance, such as accuracy and recall, will result in different results with various data. Consequently, it is hard to evaluate the quality of the systems with these metrics.

In this paper, we mainly investigate how to evaluate the quality of text localization systems. They take an image as input and report the positions of text segments in the form of boxes. An image may contain multiple text segments. The outputs provided by a text localization system may not be complete. For example, if the clarity of some text segment in the image is low, it may not be located. We adopt *stability* as the measurement, i.e., a stable system should output consistent results with similar inputs. Intuitively, the stability of a system with higher precision may be better than that of another system with lower precision. However, if a poorly implemented system cannot detect anything, it is still stable. Hence, we cannot guarantee that the precision of a stable system is high. Meanwhile, as a text localization system is sensitive to the quality of images, the valuation of the stability metric may be different with various inputs. Nevertheless, such sensitivity may guide the diagnosis of the given system.

The robustness of deep neural networks against a fixed disturbance can be formally checked by encoding the disturbance and the structures of networks with SMT (Satisfiability Modulo Theories), MILP (Mixed-integer linear programming), or abstract interpretation techniques (Huang et al., 2020). However, it cannot be widely adopted for the scalability problem. Though testing cannot provide a quantitative answer on the robustness degree of a system, it is one of the efficient ways to reveal system failures. Unfortunately, one failure of a learning-based system may not imply a bug in the implementation, for the failure may be caused by the uncertainty of the learning technique. Therefore, a small number of testing activities is unconvincing to evaluate system quality and performance. Multiple methodologies and techniques, and sufficient test cases are required to analyze a system from different aspects. Metamorphic testing is one of

[☆] Editor: Raffaela Mirandola.
[*] Corresponding author at: State Key Laboratory of Computer Science, ISCAS, China.
E-mail addresses: yrj@ios.ac.cn (R. Yan), yanjun@ios.ac.cn (J. Yan).

https://doi.org/10.1016/j.jss.2021.111040
0164-1212/© 2021 Elsevier Inc. All rights reserved.

the popular black-box testing techniques (Chen et al., 2018). Without any oracles, it can detect problems by comparing the relations between specified inputs and outputs, and be applied to various domains, such as search engines (Zhou et al., 2016) and self-driving systems (Zhou and Sun, 2019).

As metamorphic relations can define the relationship between original inputs and their variations for text localization systems without the effort of labeling outputs, we can transform the original test cases with specific metamorphic relations to synthesize a series of new cases. With the original and the synthesized test cases, we then adopt the metamorphic testing technique to check the expected relations between the outputs w.r.t. their inputs, thus evaluating system stability. Ideally, for a stable system, the introduced transformation should not lead to inconsistent results. When facing certain transformations on an image, such as brightness change or color switch, an unstable system may provide outputs different from those of the original.

The contributions of the paper are multi-fold. First, we provide an effective method to evaluate the stability of text localization systems. The difficulty in analyzing such systems is the oracle problem, for the outputs are influenced by various factors such as position, angle, or clarity of texts in an image. We propose to adopt the metamorphic testing technique to check the stability of such systems. Second, by considering potential realistic transformations on inputs, we reuse some existing metamorphic relations and put forwards problem-specific ones between inputs and outputs for text localization systems, involving both linear and non-linear relations. We tailor the synthesis step to avoid generating unrealistic images thus resulting in unfair conclusions. Third, we conduct extensive experimentation on both academic and commercial text localization tools, to evaluate system stability. We find that all the considered tools may provide inconsistent outputs for inputs with the same text segments. Meanwhile, they exhibit various sensitivity to different relations. Fourth, with the results obtained from stability analysis, we further explore the connection between the features of images and the metamorphic relations, to guide the diagnosis of text localization systems.

The rest of the paper is organized as follows. We recap the concepts on text localization systems and define the metrics for similarity comparison between two outputs, and review the definitions on metamorphic testing in Section 2. Section 3 presents the workflow of our methodology and defines metamorphic relations for text localization systems. Section 4 demonstrates experimental results. Sections 5 and 6 discuss threats to validate the method and the related work. Finally, Section 7 concludes.

2. Preliminaries

We first introduce concepts and notations for text localization systems, then recap the concepts of metamorphic testing.

2.1. Text localization systems

A text localization system aims to determine the positions of text segments in an image. The position is often represented with four points to show the corners of the text segments. The shape of the target bounding box may be rectangle or quadrilateral.

An image consists of a set of pixels identified with a two-dimensional position and a color in RGB color space. Let α be an image consisting of a set of pixels $\{\theta_i = (p, col)\}_{1 \leq i \leq n}$, where the position of pixel θ_i is denoted by $\theta_i.p = (x, y)$ and its RGB color is denoted by $\theta_i.col = (r, g, b)$. Let $B = f(\alpha)$ be the detected text segments from image α by a text localization system f, where $B = \{b_j = (p_1, p_2, p_3, p_4) \mid 1 \leq j \leq m\}$ is a finite set of boxes bounded by the positions of four points.

The text segments in an image usually consist of the contents and the positions. However, the text localization systems only report the positions of texts existing in the input. This fact allows us to define a *simple* semantics of the inputs, which only consider the position of text segments and ignore the contents in this paper. That is, given two images α_1 and α_2, if the positions of the texts in the images are the same, we say that the images α_1 and α_2 have the same *semantics*.

For the images with the same semantics, the localized positions of text segments from an unstable system may be different. To evaluate the similarity degree between the two boxes, we introduce the notion of *similarity* on the detected boxes. Let $region(b)$ be the region occupied by box b. Given two boxes b_i and b_j, if the ratio of their overlapped regions to the union of the regions is large enough, we regard the two boxes as matched. Given the tolerable error ϵ and two boxes, function *match* is defined as follows:

$$match(b_i, b_j) = \begin{cases} true & similar(b_i, b_j) > 1 - \epsilon \\ false & similar(b_i, b_j) \leq 1 - \epsilon \end{cases} \tag{1}$$

where

$$similar(b_i, b_j) = \frac{|region(b_i) \cap region(b_j)|}{|region(b_i) \cup region(b_j)|} \tag{2}$$

In Eq. (2), $|region(b)|$ denotes the size of $region(b)$. When the occupied regions of two boxes from two images are similar, they are matched. With Eq. (1), we can further reason the relationship between two sets of detected boxes.

For two sets of boxes B_1 and B_2, the set of matched boxes between the two sets is

$$M_{B_1,B_2} = \{b \in B_1 | \exists b' \in B_2.match(b, b')\} \tag{3}$$

If no redundant boxes cover the same texts, we have $|M_{B_1,B_2}| = |M_{B_2,B_1}|$. When two sets of boxes are the same, we have $B_1 = B_2 = M_{B_1,B_2}$. With the above equation, we can evaluate the similarity of two sets of boxes by

$$\delta(B_1, B_2) = \begin{cases} \dfrac{|M_{B_1,B_2}|}{|B_1| + |B_2| - |M_{B_1,B_2}|} & \text{if } B_1 \neq \emptyset \vee B_2 \neq \emptyset \\ 1.0 & \text{if } B_1 = \emptyset \wedge B_2 = \emptyset \end{cases} \tag{4}$$

Eq. (4) explains that when two sets of boxes are not empty, we can calculate their similarity according to the matched boxes. However, if both sets of boxes are empty, the similarity is one. Ideally, a text localization system f should provide the same outputs for two images α_1 and α_2 with the same semantics, i.e., $\delta(f(\alpha_1), f(\alpha_2)) = 1.0$, and we say that the results of $f(\alpha_1)$ and $f(\alpha_2)$ are *consistent*.

2.2. Metamorphic relation and testing

Informally speaking, if a system is correctly implemented, and the inputs of a system satisfy some relation, the outputs of the corresponding inputs should also satisfy a related relation. Such relations are the necessary properties of the system, and are called metamorphic relations. In other words, a metamorphic relation is any relation among the inputs and the outputs of multiple executions of a function or system. Next, we provide the formal definition of metamorphic relation.

Definition 1 (*Metamorphic Relation (MR) (Chen et al., 2018)*)**.** Let g be a target function or algorithm. A *metamorphic relation* of g is over a sequence of two or more inputs $\langle x_1, \ldots, x_n\rangle$ where $n \geq 2$, and their corresponding outputs $\langle g(x_1), \ldots, g(x_n)\rangle$. It can be expressed as a relation $\mathcal{R} \subseteq X^n \times Y^n$, where X^n and Y^n are the Cartesian products of n input and n output spaces, respectively.

The relations can be extracted from the properties of the target function and the characteristics of the input.

Given an MR $\mathcal{R}(x_1, \ldots, x_n, g(x_1), \ldots, g(x_n))$, let x_j $(k + 1 \leq j \leq n)$ be the inputs constructed based on $\mathcal{R}(x_1, \ldots, x_k, g(x_1), \ldots, g(x_k))$ over $\mathcal{R}$. We call x_i with $1 \leq i \leq k$ a *source input* and x_j with $k + 1 \leq j \leq n$ a *follow-up input*. A source input may generate multiple follow-up inputs.

If an implementation conforms with the given specification, the metamorphic relations should be preserved for all the inputs. Consider an implementation g, and the metamorphic relation $\mathcal{R}$ over implementation g. The steps for metamorphic testing of implementation g with $\mathcal{R}$ are as follows.

- Execute a sequence of source inputs $\langle x_1, \ldots, x_k \rangle$, and obtain the outputs $\langle g(x_1), \ldots, g(x_k) \rangle$. Construct and execute a sequence of follow-up inputs $\langle x_{k+1}, \ldots, x_n \rangle$ according to $\mathcal{R}$ and obtain their respective outputs $\langle g(x_{k+1}), \ldots, g(x_n) \rangle$.
- Check the results regarding relation $\mathcal{R}$. If relation $\mathcal{R}$ is not satisfied, the metamorphic relation $\mathcal{R}$ reveals that implementation g is faulty.

3. Stability evaluation via metamorphic testing

Many state-of-the-art text localization systems are implemented with learning-based techniques, especially with deep neural networks, to improve the performance for localizing text segments. Testing deep neural networks usually requires labeled outputs to check whether the actual outputs are the same as expected. However, labeling is an all-consuming process, especially for text localization systems that contain many text segments in the inputs. Apart from the quantitative evaluation of system robustness or precision, we concentrate on stability evaluation, i.e., whether a text localization system will always provide consistent outputs for images with the same semantics. Metamorphic testing can avoid test oracle problems by checking whether the outputs preserve certain relations w.r.t. the specified relations between the inputs. Due to this advantage, we adopt the technique to check the stability of text localization systems.

We illustrate the steps of our methodology in Fig. 1. We first define the metamorphic relations w.r.t. the properties of a text localization system and the features of the input. Second, we synthesize the follow-up inputs according to the defined relations, the set of source inputs, and the corresponding outputs. Third, the follow-up inputs are fed into the system, and we obtain the corresponding results. Finally, we collect the results from the source and follow-up inputs, and check whether they can preserve the metamorphic relations. The feedback from source results to data synthesis in Fig. 1 shows that synthesizing follow-ups may need the outputs of sources.

When there is no confusion, in the rest of the paper we simplify source input and its follow-up inputs as source and follow-ups, respectively.

3.1. The introduced metamorphic relations

Informally speaking, the essential property of a text localization system is that the system should extract all the positions of text segments in an image. Consequently, the system should provide consistent outputs w.r.t. inputs with the same semantics. As images are the inputs of text localization systems, the semantics of images should keep the same after reasonable transformation such as color switches or limited brightness variations. Meanwhile, if we add or delete some text segments, the semantics for the unmodified parts should be the same. Concerning these features, we adopt a set of existing metamorphic relations for images, and propose three metamorphic relations tailored for text localization systems. The metamorphic relations can be organized into two groups according to the influence on the semantics of images:

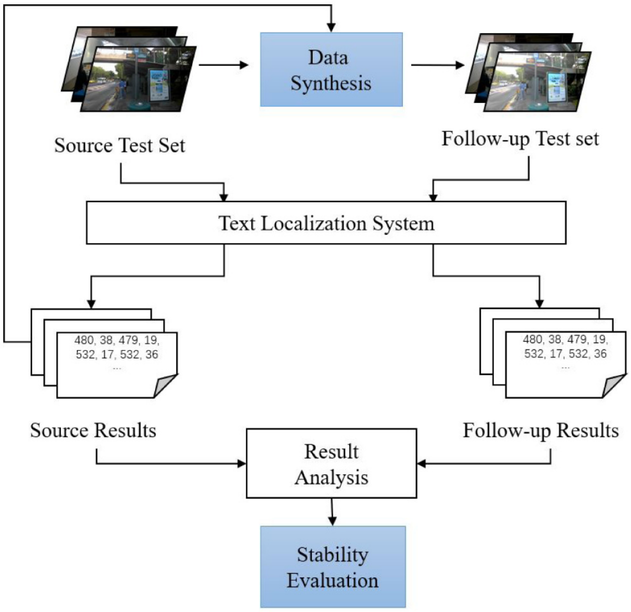


Fig. 1. The steps of our stability evaluation method.

- semantics are preserved:
  - increasing brightness (MR_{ib}): when we create a follow-up by increasing the brightness of the source within a certain limit, the outputs from the source and the follow-up should be the same.
  - decreasing brightness (MR_{db}): when we create a follow-up by decreasing the brightness of the source within a certain limit, the outputs from the source and the follow-up should be the same.
  - channel switch (MR_{cs}): when we create a follow-up by switching the RGB channel of the source, the outputs from the source and the follow-up should be the same.
- semantics are not preserved:
  - perspective transformation (MR_{pt}): when we create a follow-up by slight distortion of the source, the number of detected text segments should be the same, but with various positions.
  - watermarking (MR_{wa}): when we create a follow-up by adding a text segment on locations without any texts (watermarking), the number of detected text segments from the follow-up should be larger than that of the source.
  - masking (MR_{ma}): when we create a follow-up by covering some text segments (masking), the number of detected segments from the follow-up should be smaller than that of the source.

The selection of these metamorphic relations is to reflect realistic problems of images. For example, when we take a photo, the brightness may be low, or the objects may be distorted. We do not consider metamorphic relations such as rotation, or scaling, for they may increase the difficulty of text detection, or lead to the change in both the number of texts and their positions.

We adopt a running example shown in Fig. 2 to explain the metamorphic relations between the source and its follow-ups. The image is on a bus stop, with text segments on the route board and the advertisement board nearby. To emphasize the detected text segments, we ignore the area without any text segments in comparing the difference between detected text segments.

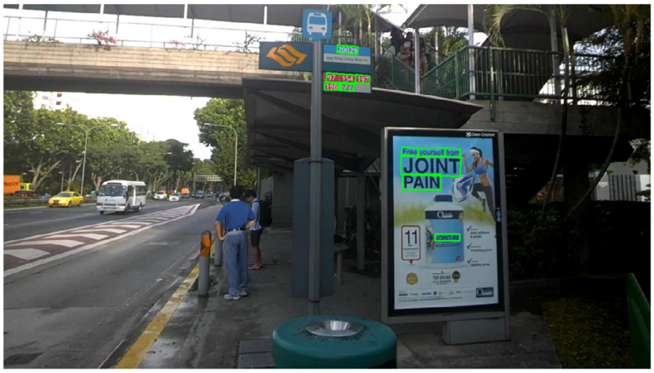


Fig. 2. The running example for an out-door image.

3.1.1. *Metamorphic relations on brightness variation*

To define the metamorphic relations on brightness variation, we first introduce the function for brightness adjustment. Given an image α, let $\kappa(x, k_1, k_2)$ be the function for brightness modification, where $k_1 > 0$, $-255 \leq k_2 \leq 255$, and

$$\kappa(x, k_1, k_2) = \begin{cases} k_1 * x + k_2 & 0 \leq k_1 * x + k_2 \leq 255 \\ 0 & k_1 * x + k_2 < 0 \\ 255 & k_1 * x + k_2 > 255 \end{cases} \tag{5}$$

As the range of RGB color is from 0 to 255, Eq. (5) ensures that the modification of a certain color will not go beyond the limit. Parameters k_1 and k_2 are called the *gain* and *bias* parameters, to control contrast ratio and brightness, respectively (Szeliski, 2011). With Eq. (5), the follow-up of source α for increasing brightness is created by

$$\begin{aligned} \gamma_{+b}(\alpha) = \ & \{(\theta.p, \theta.col') \mid \theta \in \alpha, \theta.col' = (\kappa(\theta.col.r, k_1, k_2), \\ & \kappa(\theta.col.g, k_1, k_2), \kappa(\theta.col.b, k_1, k_2)), \\ & k_1 > 0, 0 \leq k_2 \leq 255\} \end{aligned} \tag{6}$$

And the follow-up of source α for decreasing brightness is created by

$$\begin{aligned} \gamma_{-b}(\alpha) = \ & \{(\theta.p, \theta.col') \mid \theta \in \alpha, \theta.col' = (\kappa(\theta.col.r, k_1, k_2), \\ & \kappa(\theta.col.g, k_1, k_2), \kappa(\theta.col.b, k_1, k_2)), \\ & k_1 > 0, -255 \leq k_2 < 0\} \end{aligned} \tag{7}$$

Eqs. (6) and (7) show that brightness modification is mainly achieved by adjusting the parameter of k_2, i.e., when k_2 is greater than or equal to zero, it is to increase the brightness. Otherwise, it is to decrease the brightness.

Given an image α, let $B_1 = f(\alpha)$ and $B_2 = f(\gamma_{+b}(\alpha))$, we should have $M_{B_1,B_2} = B_1 = B_2$, and $\delta(B_1, B_2) = 1.0$.

We present the running example with the two metamorphic relations in Fig. 3. The image in Fig. 3(a) is the source together with the localized texts in green boxes from a text localization tool. The images in Figs. 3(b) and 3(c) are the follow-ups together with the localized texts by increasing and decreasing brightness, respectively. Among the three images, we observe that an additional text segment in Figs. 3(b) and 3(c) is detected respectively in the top right corner of the advertisement board, compared with the output of the source, which shows the inconsistency between their outputs.

3.1.2. *Metamorphic relations on switching RGB channels*

As mentioned earlier, an image contains three channels (RGB). When we permute the channels, the color on the texts and the background changes simultaneously. Therefore, the colors of the texts and the background will not be the same. And the semantics of an image should be the same before and after the channel switch. Since an image contains three channels, there are six ways of permutation. Given an image α, the way of constructing follow-ups is defined by:

$$\begin{aligned} \gamma_s(\alpha) = \ & \{(\theta.p, \theta.col') \mid \theta \in \alpha, \\ & \theta.col' = \sigma(\theta.col.r, \theta.col.b, \theta.col.g)\} \end{aligned} \tag{8}$$

where $\sigma(c_1, c_2, c_3)$ is a non-trivial permutation function, i.e., $\sigma \in \{(c_1c_2c_3), (c_1c_3c_2), (c_1c_3), (c_1c_2), (c_2c_3)\}$, in which $(c_1c_2c_3)$ is the permutation from (c_1, c_2, c_3) to (c_2, c_3, c_1), and (c_1c_3) is the permutation from (c_1, c_2, c_3) to (c_3, c_2, c_1). Given a text localization system f, an image α, and $\alpha' = \gamma_s(\alpha)$, we should have $M_{f(\alpha),f(\alpha')} = f(\alpha) = f(\alpha')$, and $\delta(f(\alpha), f(\alpha'))$=1.0.

Reconsider the running example shown in Fig. 4. The source in Fig. 4(a) is with channel sequence BGR. The follow-up in Fig. 4(b) is with channel sequence GRB, and the other in Fig. 4(c) is with channel sequence RBG. Compared with the results shown in Fig. 4(a), we observe that in the top left corner of Fig. 4(b), a text segment detected in the source is split into two. For the follow-up in Fig. 4(c), an additional text segment is detected in the top right corner of the advertisement board, which is different from the results shown in Figs. 4(a) and 4(b).

3.1.3. *Metamorphic relation on perspective transformation*

To simulate a slight distortion w.r.t. the source, we can randomly create a new image whose text segments are the same, but the shape is irregular, compared with the source. We then transform the new image in irregular shape into one in the rectangle by filling up the new areas in black. In this case, the size of a follow-up is larger than its source.

As we do not expect that the text segments in the generated image become unrecognizable, the follow-up is synthesized by randomly stretching the source from its four corners with a small degree. Once the stretching degree is decided, we can obtain a 3 × 3 matrix calculated from perspective transformation, to measure the offset of every pixel from the source to the synthesized image. The matrix is denoted by

$$\omega = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{bmatrix}$$

The pixels in an image are two-dimensional. However, the perspective transformation considers three dimensions. For convenience, we append an additional dimension z in the position of a pixel $\theta.p$ and assign it by 1, and the augmented position is denoted by $\hat{p} = aug(p) = (x, y, 1)$. Then the augmented position of pixel $\theta.\hat{p} = (x, y, 1)$ in the source will be transformed into that in the follow-up by $\hat{p}' = (x', y', z') = perspective(\hat{p}, \omega)$, where $\hat{p}'$ is the augmented position of pixel θ after perspective transformation, with

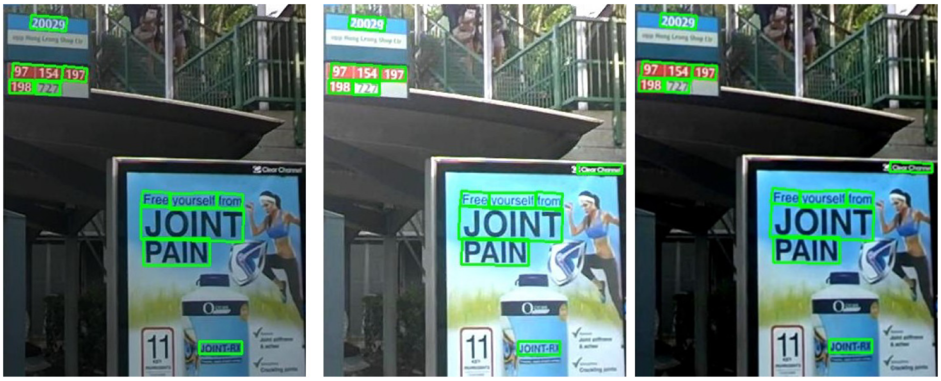
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \omega \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{9}$$

To recover the position from a three-dimensional coordinate to a two-dimensional, we need to set z' to be 1 again. Therefore, the position of θ is normalized to $(x'/z', y'/z', 1)$.

With the above equation, the follow-up of an image α with perspective transformation ω is

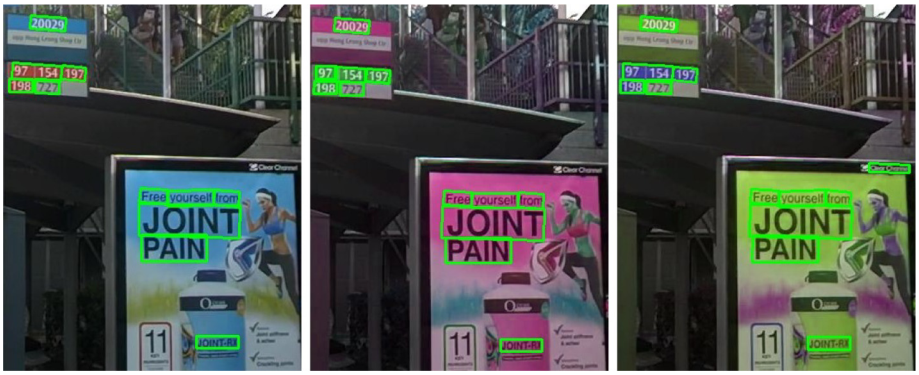
$$\begin{aligned} \gamma_p(\alpha, \omega) = \ & \{(\theta.p', \theta.col) \mid \theta \in \alpha, \\ & (x', y', z') = perspective(aug(\theta.p), \omega), \\ & p' = (x'/z', y'/z')\} \end{aligned} \tag{10}$$

Given an image α, and follow-up $\alpha' = \gamma_p(\alpha, \omega)$, we should have $|f(\alpha)| = |f(\alpha')|$, and for any box $b = (p_1, p_2, p_3, p_4) \in f(\alpha)$,



(a) source (b) follow-up ($k_2 = 30$) (c) follow-up ($k_2 = -30$)

Fig. 3. Comparison with brightness variation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) source (b) follow-up (GRB) (c) follow-up (RBG)

Fig. 4. Comparison on channel switch.

there should exist a box $b' = (p'_1, p'_2, p'_3, p'_4) \in f(\alpha')$, such that $p'_i = perspective(aug(p_i), \omega)$ for $1 \leq i \leq 4$. That is, the source and the follow-up should contain the same number of text segments, and the locations of the segments should preserve the perspective transformation relation.

We present the comparison on perspective transformation in Fig. 5 for the running example, where the size of the image in Fig. 5(b) is larger than that in Fig. 5(a). Concretely, the size of the follow-up is $1380 * 820$, and the size of the source is $1280 * 720$. Compared with the two groups of detected segments in the two images, we can observe that a route number on the route board of buses is not detected in the follow-up.

3.1.4. Metamorphic relation on watermarking

Watermarking is to integrate some identification into a digital carrier without affecting the original. For text localization systems, when an image is watermarked, the number of detected text segments should not be smaller than before. We do not expect that the original text segments are covered by any introduced watermarks, which may cause problematic text localization. Therefore, the locations of existing text segments should be detected and avoided in watermarking process. Given a string of text *str* and an image α, the synthesis with watermarking is defined by

$$\gamma_w(\alpha, f(\alpha), str) = addtext(\alpha \setminus \bigcup_{b \in f(\alpha)} region(b), str) \tag{11}$$

where *addtext*() is the function for watermarking, and $\alpha \setminus \bigcup_{b \in f(\alpha)} region(b)$ is to ensure that the introduced watermark will not be in the already occupied regions.

Given an image α and a string *str*, let $\alpha' = \gamma_w(\alpha, f(\alpha), str)$. Then we should have $f(\alpha) \subset f(\alpha')$ and $|f(\alpha')| = |f(\alpha)| + 1$.

Reconsider the running example with watermarking shown in Fig. 6. The image synthesized by randomly adding the watermark beyond the occupied regions is in Fig. 6(a). The localized text segments of the follow-up are shown in Fig. 6(b). We find that both the watermark and previously detected text segments are detected from the follow-up.

3.1.5. Metamorphic relation on masking

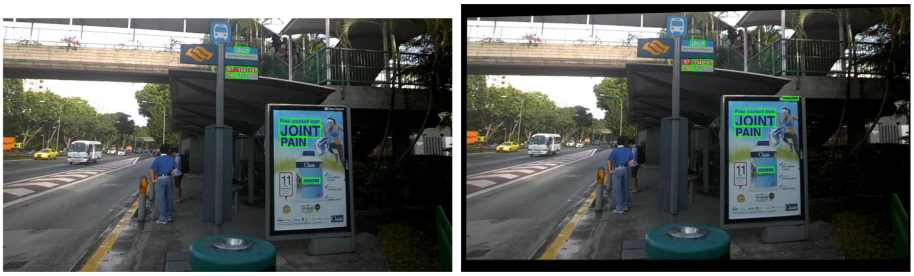
When the texts in an image are covered and unrecognizable, they should not be detected and localized anymore. Constructing follow-ups with such a relation involves two steps. First, the text segments are detected. Second, the detected regions are masked by changing the colors of pixels in the detected region such that the texts are not recognizable. Here, we consider masking all the texts by changing the color of the pixels with the background of the boxes. Given an image α, the follow-up is synthesized as follows:

$$\gamma_m(\alpha, f(\alpha)) = \{(\theta.p, \theta.col') \mid \theta \in \alpha, \theta.col' = mask(\theta, f(\alpha))\} \tag{12}$$

where

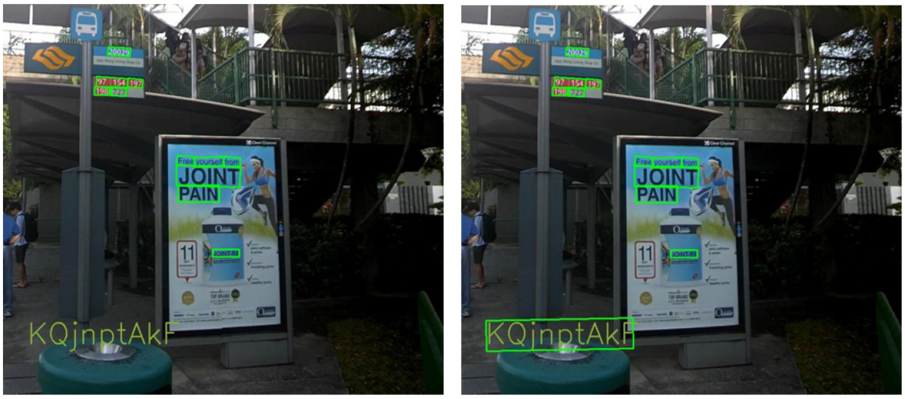
$$mask(\theta, f(\alpha)) = \begin{cases} \theta.col & \text{if } \forall b \in f(\alpha).\, \theta.p \notin b \\ b.p_1.col & \text{if } \exists b \in f(\alpha).\, \theta.p \in b \end{cases} \tag{13}$$

Eq. (13) is to change the color of pixels in a text box with the color of the pixel in the top left corner of the box. And the pixels outside any detected text boxes keep the same. For an image α and its follow-up $\alpha' = \gamma_m(\alpha, f(\alpha))$, we should have $f(\alpha') = \emptyset$.



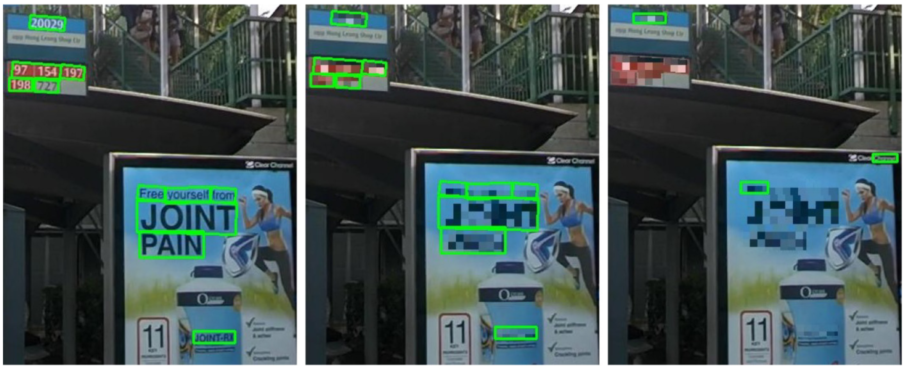
(a) source (b) follow-up

Fig. 5. Comparison on perspective transformation.



(a) follow-up (b) follow-up results

Fig. 6. Comparison on watermarking.



(a) source (b) follow-up (c) follow-up results

Fig. 7. Comparison on masking.

Consider the running example shown in Fig. 7. We show the detected text boxes from the source in Fig. 7(a). The masked follow-up and the detected boxes from the follow-up are shown in Figs. 7(b) and 7(c), respectively. As we have masked all the identified texts in the follow-up, the tool should not detect any boxes. However, two previously identified boxes are still detected, and a new text segment is detected in the top right corner of the advertisement board.

3.2. Metrics for stability evaluation

The stability of a text localization system is evaluated with the inconsistency between images with similar semantics. The similarity metric provides a way to measure inconsistency degree. Therefore, we mainly adopt similarity as the basis to evaluate system stability. A source may contain multiple follow-ups. The metric defined in Eq. (4) is to evaluate the similarity between boxes from a pair of inputs. To evaluate the similarity between a source and its follow-ups, we extend the equation for evaluating the average similarity between the outputs from a set of source and follow-up inputs.

Given an image α and a metamorphic relation γ, let $I_{\gamma_\alpha} = \{\alpha' \mid \alpha' \text{ is a follow-up input of source } \alpha \text{ w.r.t. relation } \gamma\}$ be the set of follow-up inputs synthesized from source α and metamorphic relation γ. The average similarity between the outputs of source α and follow-ups in I_{γ_α} over system f is defined as follows:

$$\delta_\gamma(\alpha) = \frac{\sum_{\alpha' \in I_{\gamma_\alpha}} \delta(f(\alpha), f(\alpha'))}{|I_{\gamma_\alpha}|} \tag{14}$$

That is, the average similarity considers the similarity between every pair of source and follow-up outputs, to evaluate the sensitivity of a system over an image with certain metamorphic relations.

Stability evaluation is usually conducted with a set of images. When we need to consider the similarity between outputs over a set of images I and their follow-ups with metamorphic relation γ, we introduce *set similarity* as follows:

$$\hat{\delta}_{\gamma}(I) = \frac{\sum_{\alpha \in I} \delta_{\gamma}(\alpha)}{|I|} \tag{15}$$

Eq. (15) evaluates the sensitivity of a system over a set of images with metamorphic relation γ. Ideally, the value of set similarity is one. The higher set similarity shows that the system is more stable.

When the source and its follow-ups have the same semantics, their outputs are supposed to be similar. Therefore, we can adopt Eq. (15) to evaluate the set similarity between images with the same semantics. However, when the semantics of the source and its follow-ups are different, the metric of set similarity cannot indicate the difference of outputs between a source and its follow-ups. We need additional metrics to evaluate the difference between the expected and the actual outputs.

For the metamorphic relation with watermarking, if more introduced watermarks are detected, the stability of the system is better. Consequently, we propose the metric of *shooting rate* for stability evaluation under the metamorphic relation with watermarking, which is the total times that watermarks are detected to the number of all follow-ups.

For the metamorphic relation with masking, the system is not stable if a masked text segment is detected. We introduce *success rate* to evaluate the rate that none of the masked text segments in an image can be detected. Let $\alpha \in I$ be the source, and $\alpha' = \gamma_m(\alpha, f(\alpha))$ be the synthesized follow-up. Let function $succ(\alpha')$ be 1 if $f(\alpha') = \emptyset$, indicating that previous text segments are not detected in the follow-up of source α, and no new text segments are detected. Otherwise $succ(\alpha')$ is 0. Then, the success rate of text segment detection over a set of images I with relation masking is

$$succeed(I) = \frac{\sum_{\alpha \in I, \alpha' = \gamma_m(\alpha, f(\alpha))} succ(\alpha')}{|I|}. \tag{16}$$

Among all the criteria, metrics with larger values indicate better performance.

4. Experimentation

In the experimentation, we mainly validate the following questions:

- Q1 (Effectiveness): Will the text localization system exhibit diverse evaluation results with the proposed MRs?
- Q2 (Redundancy): Are the proposed MRs redundant?
- Q3 (Usability): Can we find the causes of low stability with the MRs and the evaluation results?

To answer question Q1, we need to check whether the valuation of the metrics for various metamorphic relations is diverse, where the metrics measure the stability of the tools w.r.t. the given inputs. If the valuations over a tool are not the same for the same image with different MRs, the MRs can effectively reveal the difference in the stability of the tool. Moreover, if various tools present different values with the same image and the same MR, the stability of the tools is different. The difference of metric valuations within a tool and between various tools demonstrates the effectiveness of the method in stability evaluation. To answer question Q2, we need to compute the correlation of stability valuations between various MRs. If the correlation between two MRs is low, none of the MRs is redundant. To answer question Q3, we need to check whether the MRs leading to the low stability of the tools can be used to improve system performance or reveal system defects.

Table 1
The performance of three academic systems with the dataset of ICDAR 2015 (ICDAR2015, 2015).

Name	Recall	Precision	F-Score	Method
PSENet	85.22%	89.30%	87.21%	segmentation
PixelLink	83.77%	86.65%	85.19%	segmentation
EAST	77.32%	84.66%	80.83%	regression

4.1. Experimental setup

The automatic synthesis of follow-ups and data analysis in the methodology is implemented in Python. The effectiveness of the methodology is evaluated with both academic and commercial text localization tools. The data of experimentation is available from https://anonymous.4open.science/r/795aeae2-4930-4d97-b4b8-b281f39cb0d0/.

We consider three academic text localization tools to investigate their stability, including PSENet (Wang et al., 2019), PixelLink (Deng et al., 2018), and EAST (Zhou et al., 2017), which can detect and report the locations of texts. The three tools are open source, and have been tested with the official test set provided by ICDAR 2015 (the 13th International Conference on Document Analysis and Recognition).[1] This fact allows us to have a fair evaluation of their stability.

Both PSENet and PixelLink are segmentation-based methods. PSENet adopts a kernel-based framework. It fulfills pixel-level segmentation, which can precisely locate the text instance with an arbitrary shape. PixelLink links and segments text instances according to the positions of pixels. EAST adopts regression-based methods and applies fully convolutional networks (FCN) (Long et al., 2015) to directly predict score maps. As the version of adopted FCN in EAST is not open-source, we consider the implementation of EAST with a 50-layer ResNet (Anon, 0000) in our experimentation, which achieves the same performance as the previous implementation.

We present their performance evaluation results with the official dataset from ICDAR 2015 in Table 1. In the table, *Recall* is the percentage between the number of detected boxes matching ground truth and the number of all ground-truth boxes; *Precision* is the percentage between the number of detected boxes matching ground truth and all detected boxes; and *F-Score* is the harmonic mean of precision and recall (Derczynski, 2016). From the table, we can observe that the performance of PSENet is the best. The performance of EAST is not as good as the other two. We use Colab[2] with GPU acceleration as the experimentation environment for these tools.

Additional to evaluating the stability of academic text localization tools, we also investigate the stability of commercial tools that can provide text localization services, including Google Cloud platform (GCP),[3] Amazon Web Service (AWS),[4] Azure (Microsoft Cloud)[5] and Tencent Cloud.[6] The details about the commercial systems are shown in Table 2. For the Google Cloud platform,

[1] http://www.iapr-tc11.org/archive/icdar2015/.

[2] https://colab.research.google.com/.

[3] https://cloud.google.com.

[4] https://docs.aws.amazon.com/rekognition/.

[5] https://www.azure.cn.

[6] https://cloud.tencent.com/product/generalocr.

Table 2
Investigated commercial systems.

Platform	API Name	Version/Last used
Google Cloud Platform	Vision AI	Apr-20
Amazon Web Services	Amazon Rekognition	Apr-20
Azure	Cognitive services	Apr-20
Tencent	GeneralAccurateOCR	v2018-11-19

Table 3
The details on follow-ups and the evaluation criteria.

MRs	Amount	Criteria	Ideal
MR_{ib}(Increasing brightness)	20*500	set similarity	1.0
MR_{db}(decreasing brightness)	20*500	set similarity	1.0
MR_{cs}(Channel switch)	2*500	set similarity	1.0
MR_{pt}(Perspective transformation)	100*500	set similarity	1.0
MR_{wm}(Watermarking)	20*500	shooting rate	1.0
MR_{ma}(Masking)	1*500	success rate	1.0

it provides API `Vision AI` to detect text within images. We upload the source and follow-up samples to Cloud Storage, run the script on Google Shell, then fetch the outputs for local analysis. For the other three platforms, i.e., Amazon Web Service, Azure, and Tencent, we upload source and follow-up inputs, then create client objects in Python script (using SDKs provided by the corresponding official platforms), call the GET API, and wait for it to retrieve the results, and fetch the outputs for local analysis.

We set ϵ as 0.5 for all the tools in identifying matched boxes, to keep the same criterion as the parameter in calculating metrics *precision* and *recall* in ICDAR 2015. In other words, when the overlapped area of two boxes is larger than the non-overlapped area, they are regarded as the same.

4.2. Original test cases and the synthesized follow-ups

In this paper, we adopt the testing dataset in ICDAR 2015 with 500 images as the set of sources, and generate the follow-ups according to the metamorphic relations defined in Section 3.

For MR_{ib} with increasing brightness, we increase the value of k_2 by 5 from 0 to 100, and obtain 20 follow-ups for one image. We obtain $20*500$ follow-ups. The case of MR_{db} is similar, which decreases the value of k_2 from 0 to -100. Though there are five possible channel switches in function γ_s, we choose the switch on GBR and GBR as the follow-ups and obtain two follow-ups for one image. For MR_{pt}, we randomly generate 100 irregular images for one image, with 100 various matrices for the distortion within a 50×50 region in the four corners of the source. For MR_{wm}, we randomly generate 20 strings whose lengths are within a certain limit, and integrate them into unoccupied regions of the source randomly. Table 3 provides the number of generated follow-ups for every metamorphic relation in column *Amount*.

We also list the metrics for evaluating the stability of text localization tools in Table 3 over the defined metamorphic relations, where column *Criteria* shows the adopted metrics and column *Ideal* provides the ideal values of the metrics. As the number of detected boxes between the outputs of sources and follow-ups should be the same for the first four metamorphic relations, system stability can be evaluated with set similarity defined in Eq. (15). However, for the metamorphic relations with watermarking and masking, the numbers of text segments in a source input and its follow-ups are different, and Eq. (15) is not applicable anymore.

4.3. Q1: Effectiveness of stability evaluation on text localization tools

To answer question Q1 and demonstrate the effectiveness of metamorphic relations in detecting inconsistent outputs over all the images, we provide the value distribution of various metrics on PSENet, and stability comparison over academic and commercial tools.

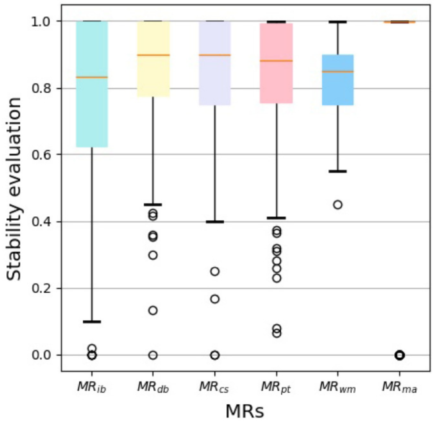


Fig. 8. The distribution of stability evaluation for PSENet.

Table 4
Stability analysis over three academic systems.

MRs	PSENet	PixelLink	EAST
MR_{ib}	0.779	0.746	0.719
MR_{db}	0.863	0.836	0.831
MR_{cs}	0.859	0.929	0.816
MR_{pt}	0.845	0.820	0.822
MR_{wm}	0.820	0.545	0.669
MR_{ma}	0.850	0.720	0.842

4.3.1. Distribution of metric valuations for PSENet

We first provide the distribution of stability evaluation for PSENet with various metamorphic relations in Fig. 8. Among the distributions from various metamorphic relations except for MR_{ma}, the distribution of metrics from MR_{ib} is the largest, showing that the tool is more sensitive to the increased brightness. We observe that the distributions between MR_{db}, MR_{cs} and MR_{pt} are similar. Though the metamorphic relation MR_{pt} does not preserve the semantics, the adjustment in follow-ups is relatively minor. Hence, the stability of the tool with MR_{pt} is similar to the cases with MR_{db} and MR_{cs}. The last two relations do not preserve the semantics of images. Accordingly, their distributions are not similar to the others. For the case with MR_{wm}, the quality of the introduced texts is better than that of the images. Consequently, most introduced texts can be detected, and the distribution of metrics is smaller. For relation MR_{ma}, if some masked texts in an image are detected, the value of the metric for the image is zero. Otherwise, the value of the metric is one. As the tool does not detect any boxes for most of the follow-ups, the Median valuation for the relation is one, and the outlier is zero.

4.3.2. Stability evaluation of academic systems

To answer question Q1, we also provide evaluation results on academic systems, showing that the metrics are effective in stability evaluation.

We list the values of metrics for the three text localization tools over the six metamorphic relations in Table 4.

First, we compare the stability of the systems over every metamorphic relation. For brightness modification, the values of metrics from increasing brightness are lower than those from decreasing brightness. This fact shows that the systems are more stable in handling images with decreased brightness. For channel switch, the stability of PixelLink is higher than the other two,

Table 5
Selected Images for Commercial Systems.

MRs	Amount	Indexes of selected images
MR_{ib}	10*10	153, 20, 207, 418, 257, 485, 88, 361, 69, 440
MR_{db}	10*10	293, 257, 129, 109, 485, 438, 21, 239, 134, 42
MR_{cs}	2*10	20, 177, 21, 109, 440, 221, 42, 187, 245, 134
MR_{pt}	10*10	109, 309, 207, 257, 386, 334, 221, 20, 253, 434
MR_{wm}	10*10	151, 20, 257, 352, 207, 253, 109, 293, 491, 177
MR_{ma}	1*30	All the indexes of selected images

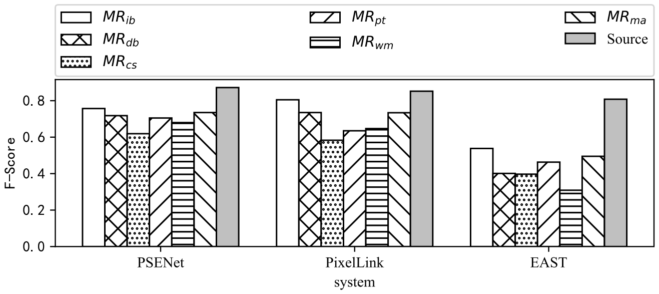


Fig. 9. F-Score comparison between selected images per MR and the origin.

Table 6
Set similarity results of selected images.

MRs	GCP	AWS	Azure	Tencent	PSENet	PixelLink	EAST
MR_{ib}	0.547	0.522	0.621	0.500	0.701	0.690	0.521
MR_{db}	0.611	0.739	0.723	0.520	0.735	0.700	0.742
MR_{cs}	0.771	0.730	0.749	0.730	**0.865**	0.848	0.694
MR_{pt}	0.677	0.728	0.676	0.632	0.683	0.693	0.646
MR_{wm}	0.630	0.800	0.770	0.720	0.580	0.450	**0.750**
MR_{ma}	0.714	0.551	0.592	0.592	0.694	0.653	**1.000**

showing that its implementation may be good at dealing with color switches. For perspective transformation, the stability of PixelLink and EAST is similar. However, for watermarking, the difference between the three tools is quite obvious. PixelLink can only detect about fifty percent of introduced watermarks. For masking, the stability of PixelLink is still not as good as the other two, showing that it has detected more masked text segments, which should not be detected.

Second, we compare the stability of the three tools over all the metamorphic relations. The metrics from PSENet always achieve the highest value except for the third metamorphic relation. Hence, it is more stable than the other two. For PixelLink, it is more stable than EAST in the case of color variation (i.e., brightness modification and channel switch). However, when the source experiences semantic modifications (with the remaining three metamorphic relations), it is not as stable as EAST.

Third, we summarize the features of the three tools based on the above analysis. All the tools are sensitive to the case with the increased brightness. PixelLink is good at dealing with color switches, but sensitive to the change of semantics. Both PixelLink and EAST are sensitive to the introduced watermarks. Except for watermarking and increasing brightness, the sensitivity of EAST to other metamorphic relations is not quite obvious.

Finally, we analyze which metamorphic relation is more efficient in revealing system defects. According to the comparison on metric valuations, increasing brightness and watermarking are more efficient than the other relations, as their valuations are lower than the others.

4.3.3. Stability evaluation of commercial systems

Additional to the evaluation of academic tools, we also investigate the stability of commercial tools.

Image selection. Some commercial tools are in the form of web services. It is difficult to check all the test cases with their follow-ups for the usage limit. We select a subset of images for this evaluation. For every metamorphic relation, the images are sorted according to the increasing order of the metric values evaluated from the three academic tools. We select the first ten images for every metamorphic relation. The intersection between sets of selected images in various metamorphic relations is not empty. We take the union of these images as the set of sources for stability evaluation on commercial tools. We list the indexes of selected sources and the number of reused follow-ups in Table 5, where the indexes of sources for the first five metamorphic relations are in the increasing order of metric valuations.

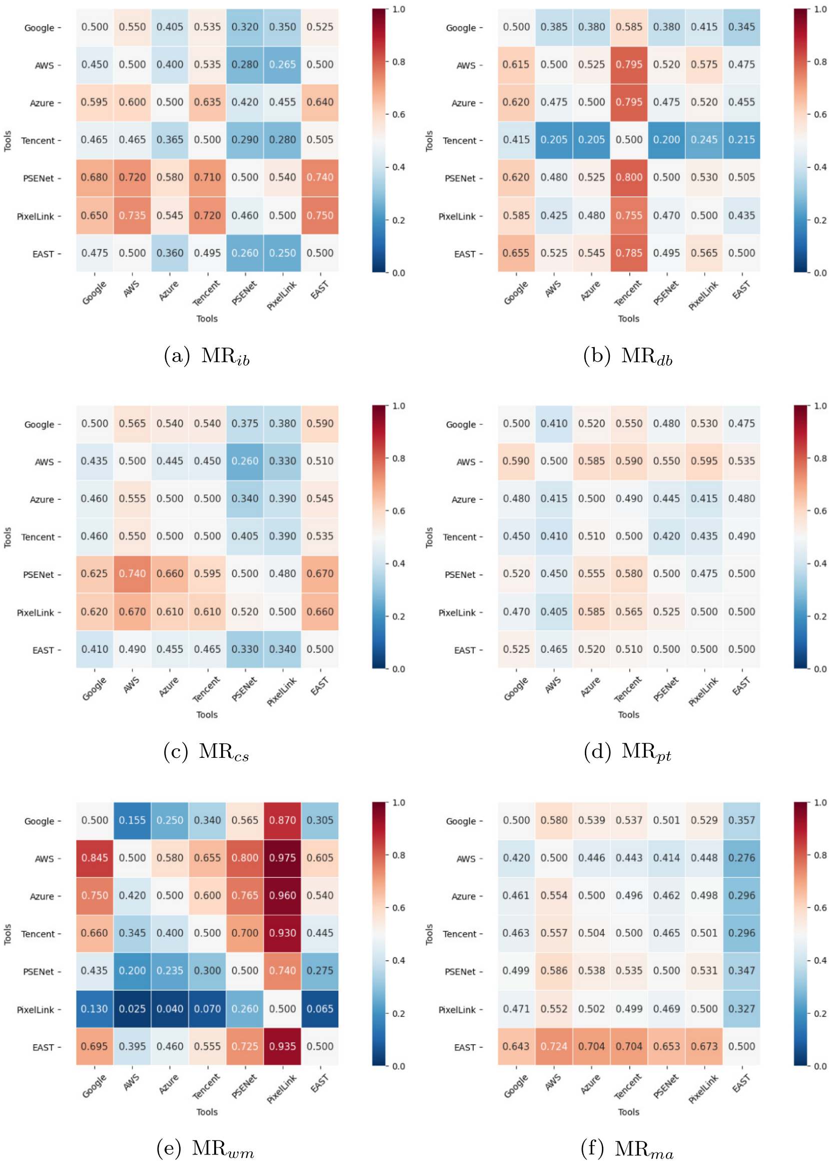
We have also tested the three academic tools with the selected images. We present the values of F-Score obtained from the subset of test cases and the synthesized images on every metamorphic relation in Fig. 9. In the figure, columns labeled with *Source* show the F-Score evaluated with the selected subset of images, and the others are those with the synthesized follow-ups from the selected images. From the figure, we observe that the values of F-Score from all the tools with the selected images are similar to those with the whole dataset. However, the values of F-Score decrease with the synthesized follow-ups. This observation validates the fact that the performance of the tools is sensitive to the inputs. Among six metamorphic relations, the difference with channel switch is the largest for PSENet. The case for PixelLink is similar. For EAST, the value of watermarking is the lowest.

Stability comparison on selected images. We present the evaluation on both commercial and academic tools with the selected images in Table 6.

We observe that the academic tools are more stable with semantic preserving metamorphic relations, and commercial tools are more stable with the watermarking relation. Among the comparison of commercial tools from Table 6, it is hard to conclude that one is always more stable than others. For example, the stability of Google Cloud platform with metamorphic relations channel switch and masking is better. Amazon Web Service is more stable with decreasing brightness, perspective transformation, and watermarking. Meanwhile, Azure performs better with increasing brightness. Moreover, the commercial tools are also sensitive to the metamorphic relation with increasing brightness.

For the comparison of academic tools with the selected images, the majority of metric valuations are lower than those from in Table 4, showing that these tools are more sensitive to the selected images. There are only three metric valuations that are higher than those in Table 4. The F-Score of EAST with the synthesized images from the masking relation is not the highest. However, the stability metric is the highest. Therefore, given a set of data, the performance of a stable system maybe not good. Meanwhile, from Table 6, we observe that PSENet is not stable for the selected inputs with watermarks. However, from Fig. 9, we find that the F-Score with the case of channel switch from PSENet is the lowest. The comparison shows that the F-Score and stability are different types of metrics.

We further apply the Friedman test (Friedman, 1937) to analyze whether the differences of the tools is statistically significant. We assume that the performance of the tools is similar. To check whether the assumption is acceptable, we set the significance level as 0.1 and the confidence interval as 0.95. The computed probabilities (p-values), which measure the evidence against the hypothesis, show that the differences with most of the metamorphic relations are significant. However, the performance



(a) MR$_{ib}$

(b) MR$_{db}$

(c) MR$_{cs}$

(d) MR$_{pt}$

(e) MR$_{wm}$

(f) MR$_{ma}$

Fig. 10. Effect sizes with Vargha–Delaney A measure.

of the tools is similar for metamorphic relations on channel switch and perspective transformation.

The effect sizes between the tools evaluated with Vargha–Delaney A measure (Vargha and Delaney, 2000) are shown in Fig. 10. For two tools A and B, if Vargha–Delaney A measure VD.A(A,B) is exactly 0.50, the two tools achieve equal performance. When the measure is less than 0.50, the performance of A is worse. When the measure is more than 0.50, the performance of B is worse. In Fig. 10(d), we observe that the effect sizes between the tools are similar, which is consistent with the results obtained from the Friedman test. In Fig. 10(c), the effect sizes between the commercial tools are closer.

We also merge Tables 4 and 6, and provide the comparison in Fig. 11, where legend *distance on all samples* on the three academic tools of every metamorphic relation shows the distance of current valuation with selected images to the original metric valuation with the whole dataset. We observe that most metric valuations on academic tools with the selected images are lower than those with the whole set. Two exceptions for EAST are more explicit, whose metric valuations on watermarking and masking with selected images are higher than those with the whole set. In Fig. 11, we also observe that the stability of commercial tools is similar for the MR with channel switch. Meanwhile, the stability of both academic and commercial tools on the selected images is similar for perspective transformation. The measures presented in Fig. 10 are consistent with the observations concluded from Fig. 11.

4.3.4. Summary

The experimental results on the distribution of stability evaluation for PSENet, and the stability comparison between various tools demonstrate that the proposed metamorphic relations are effective in detecting the inconsistent outputs, and evaluating the stability of the text localization tools.

4.4. Q2: Redundancy of metamorphic relations

To answer question Q2, we analyze the correlation with the Pearson correlation coefficient (Benesty et al., 2009) between the metamorphic relations based on the results obtained from PSENet. For every image, the calculation takes the valuation of metrics for every two metamorphic relations. The correlation

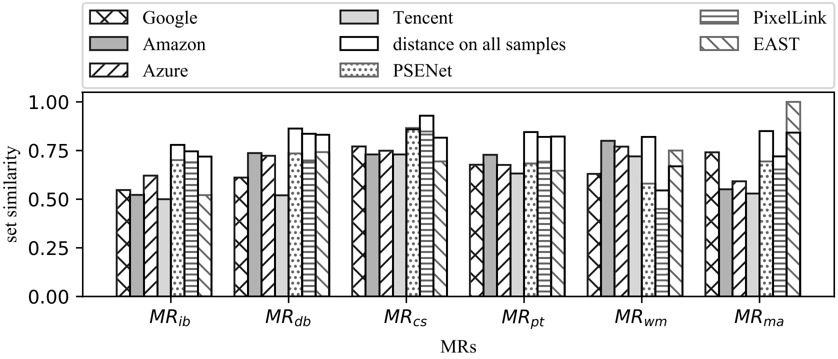


Fig. 11. Stability comparison with selected images..

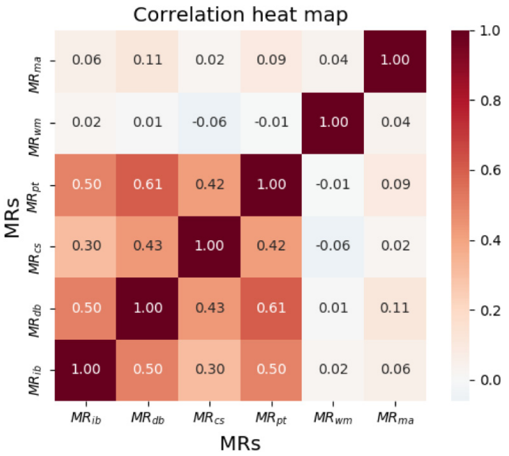


Fig. 12. Correlation between various MRs from results of PSENet.

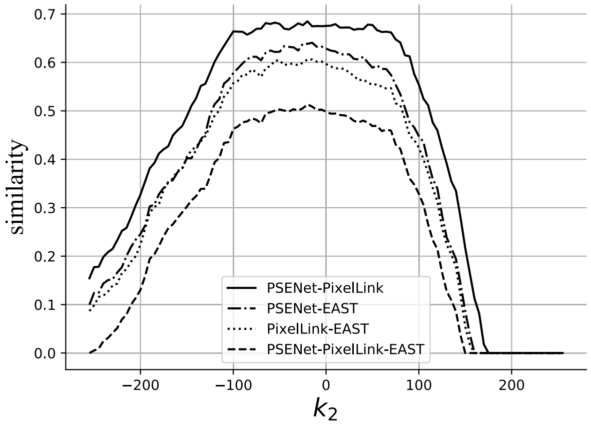


Fig. 13. The relation between brightness and stability.

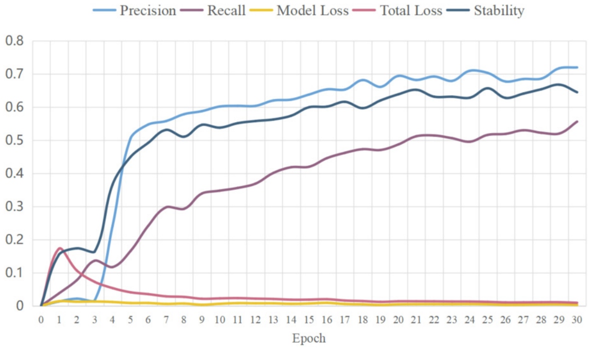


Fig. 14. The trends of precision, recall and stability in the retraining process.

between the two metamorphic relations is the average coefficient of all the images. We present the correlations between various metamorphic relations in Fig. 12.

We observe that all the values of correlations in Fig. 12 are no more than 0.61. Therefore, the metamorphic relations are not strongly related, and none of the metamorphic relations is redundant. Among the computed coefficients, the first four are higher. The reason is that the first three preserve the semantics of images, and the adjustment in perspective transformation is small. As the other two do not preserve the semantics of images, their correlations to the first four are low. The fact that the correlation between MR$_{ib}$ and MR$_{db}$ is not high is beyond our original expectation, as increasing brightness seems to be a dual of decreasing brightness. The observation leads to further investigation on the brightness modification in Section 4.5.

4.5. Q3: Usability of evaluation results

To answer question Q3 and guide the diagnosis of the tools, we have conducted a series of analyses to explore the relationship between the adopted metamorphic relations, the quality of images, and the obtained metrics.

4.5.1. Investigation on brightness modification

In Table 4, we can observe that the text localization tools are more sensitive to increasing brightness. To check the sensitivity degree of the tools to brightness, we further compare the stability of the academic tools over brightness modification. The considered range of k_2 is from -200 to 200. And the value is increased by 5 to generate a follow-up. The trend of stability comparison between the three tools over brightness modification is shown in Fig. 13, where the curves with $k_2 \geq 0$ are for increasing brightness, and the curves with $k_2 < 0$ are for decreasing brightness. The trends of the stability changing between these tools are similar. When the absolute value of k_2 is greater than 100, it becomes difficult to detect all the text segments. And the metric valuations decrease quickly. We observe that decreasing the value of k_2 from 0 to -100 has less influence on the stability of the tools. However, when $k_2 > 75$, the stability of the three tools deteriorates rapidly.

We further investigate the average brightness (Bezryadin et al., 2007) of the adopted dataset, which is also applied as the training and testing sets for the three tools. The analysis shows that the average brightness for the training set of the dataset from ICDAR 2015 is 91.19, which is lower than the average brightness

(114.86) for the training set of ICADR 2013.[7] Consequently, the tools are more stable with decreasing brightness.

We then apply the synthesized follow-ups with increased brightness to retrain PSENet. The accuracy of the model converges after 30 epochs. We present the trends of stability and the traditional metrics such as precision and recall during the retraining process in Fig. 14. We observe that the trends of recall and stability are similar. During the first three epochs, the value of stability is higher than that of precision. After five epochs, the difference between stability and precision becomes smaller. Hence, increasing system stability can improve the traditional metrics.

4.5.2. Investigation on multiple factors

Additional to brightness, we observe that the three academic tools are also sensitive to the introduced watermarks. The clarity of the introduced watermarks is high. Meanwhile, the introduced watermarks also increase the number of text segments. We explore the relation between the clarity of text segments, the number of text boxes, and the stability of PSENet.

Motion or out-of-focus blur can lead to poor clarity of images. As the two types of blur can be evaluated with the singular value decomposition algorithm (Xiao et al., 2019), we first compute the metric for every text segment, and then compute the average between all the text segments in the evaluation. The distributions of the stability metrics from the six metamorphic relations are illustrated in Fig. 15, where we regularize the clarity evaluated with a singular value decomposition algorithm and the number of text boxes. We find that there is no explicit relation between the clarity of images and the stability metrics. However, the distributions in Figs. 15(a)–15(d) demonstrate that the tool is more stable when the number of text boxes in an image is small, even if the clarity of the image is low. As the tool cannot detect any texts for most of the synthesized follow-ups from relation MR_{ma}, the density is high in the right corner of Fig. 15(f).

To investigate the relation between the stability of the tools and the clarity of text segments, we also attempt to extract the relation with linear or ridge regression analysis (Chatterjee and Hadi, 2015). However, the resulted confidence is quite low. We further adopt the Apriori algorithm (Agarwal et al., 1994) in association rule mining tool WEKA[8] to automatically extract the rules. In the mining process, we provide the tool with the values of clarity and the number of text boxes for every image, and the obtained metric valuation from stability analysis for rule extraction. According to the reported results, we know that (1) when the stability of PSENet with relation MR_{ma} is low (less than 0.33), the clarity of images is low (less than 0.50), and the confidence is 0.88; and (2) when the stabilities of PSENet with relations MR_{ib}, MR_{pt} and MR_{wm} are low (less than 0.83, 0.88 and 0.83 respectively), the clarity of images is also low (less than 0.33) with confidence 0.70. The analysis shows that when the tool is less stable, the clarity of the images is not high.

The investigation shows that the number and the clarity of text segments may influence the stability of PSENet.

4.5.3. Summary

The stability evaluation with metamorphic relations on brightness modification guides the analysis over the brightness of the images in the dataset. From the analysis, we conclude that the quality of the training set affects the performance of the tools. Though we cannot find explicit relation between the number of text segments, the clarity of the images, and the stability metrics, their distribution comparison shows that PSENet may be more stable when the number of text segments in the input is small. Therefore, the evaluation results facilitate the diagnosis of the tools.

5. Threats to validity

In this section, we discuss the external validity and internal validity of our stability evaluation method.

5.1. Threats to internal validity

In the experimentation, we adopt the well-known dataset from ICDAR 2015 to evaluate the text localization tools. To avoid deviating from realistic problems, we carefully select the metamorphic relations and tailor the synthesis process such that the evaluation is not biased. Due to the limited usage of commercial tools, we select a subset of images from the dataset. To reach a fair comparison, we evaluate both academic and commercial tools on the selected images. As the comparison has the same baseline, we can still figure out the stability of the tools under various metamorphic relations. In the investigation on the cause of failures, we also consider all the images of the dataset, to obtain a fair conclusion from the detailed analysis.

However, similar to other evaluation metrics, the stability evaluation results may vary w.r.t. the selection of images. For example, the stability metrics obtained from the images with many text segments may be different from those obtained from the images with a few text segments. Meanwhile, the stability metrics may be various from the images with various brightness or clarity. The method evaluates stability by considering the average of the metrics from a given set of images. The evaluation metrics with various groups of images may be different. As we adopt six metamorphic relations to evaluate system stability in various views, the method is still feasible.

5.2. Threats to external validity

Our method is blackbox and independent from the implementation of text localization tools. The experimentation on both academic and commercial systems shows that the method can be generalized to evaluate the stability of other text localization systems. The method only considers the positions of the texts. Hence, it can also be extended to evaluate the stability of other image-oriented object localization systems.

The dataset selected in this paper contains many indoor images with blurred backgrounds and artificial lights. However, the outdoor images still reveal inconsistent outputs from various tools. Therefore, the method can also be generalized to other datasets.

6. Related work

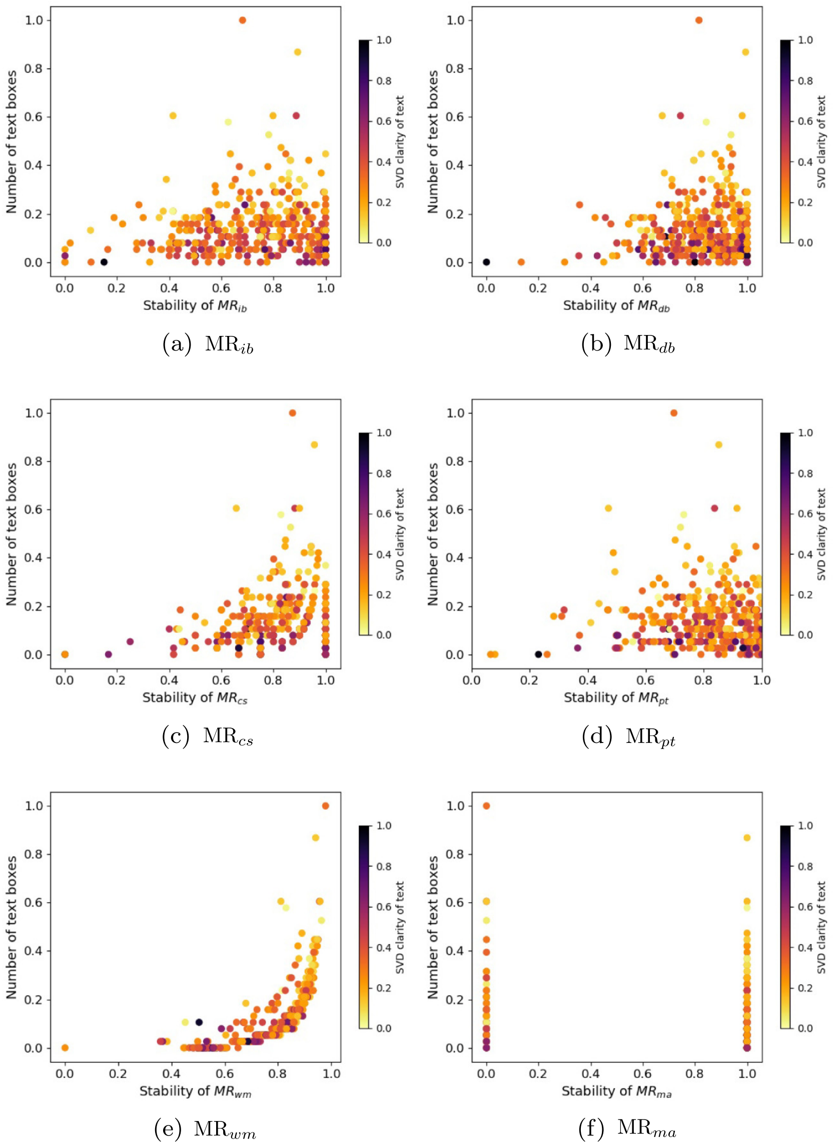
Various techniques for verification and validation of traditional systems have been adjusted to analyze the properties and performance of learning-based systems. For example, we can encode the system structure with either SMT or MILP constraints or abstract interpretation techniques to verify safety (Schwalbe and Schels, 2020) or robustness (Gehr et al., 2018). Meanwhile, testing techniques, such as coverage metrics (Pei et al., 2017) on the models or metamorphic testing techniques (Xie et al., 2009; Zhang et al., 2020), are also widely applied to check failures for learning-based systems.

Among various techniques, metamorphic testing is to reveal system defects by checking whether certain metamorphic relations are preserved between outputs from specified inputs.

Facing the difficulty of obtaining test oracles, Murphy et al. propose a series of metamorphic relations that can be applied to test machine learning applications (Murphy et al., 2009). Xie et al. design specific metamorphic relations for testing classification algorithms such as the Naive-Bayes classifier (Xie et al., 2009).

[7] https://iapr.org/archives/icdar2013/.

[8] https://www.cs.waikato.ac.nz/ml/weka/index.html.



(a) MR_{ib}

(b) MR_{db}

(c) MR_{cs}

(d) MR_{pt}

(e) MR_{wm}

(f) MR_{ma}

Fig. 15. Distributions between stability of PSENet, number of texts and clarity.

The technique has also been adopted to identify implementation bugs in learning-based image classifiers (Dwarakanath et al., 2018), whose empirical study shows that 71% of implementation bugs were caught by the introduced metamorphic relations. Additional to channel switch, the paper provides other metamorphic relations involving the manipulation of data sets and internal operations. Wang and Su synthesize images by inserting objects related to the background to test learning-based object detectors, and apply the synthesized images that can reveal detection errors for system retraining (Wang and Su, 2020).

Meanwhile, metamorphic testing is also applied to detect failures for domain-specific applications, especially for self-driving cars. For example, generative adversarial networks and metamorphic testing are adopted to validate input samples and detect behavioral inconsistencies in autonomous driving systems based on deep neural networks (Zhang et al., 2018), whose metamorphic relation consists of weather adjustments. Tian et al. apply metamorphic relations between multiple executions of systems to automatically detect the erroneous behaviors (Tian et al., 2018), whose metamorphic relations consist of scaling, rotation, brightness and contrast modification, and blurring of images. By assuming that the identified obstacles in a source should be a subset of those in a follow-up with additional LiDAR data points, previously unknown fatal errors in the obstacle perception system of a self-driving platform are detected (Zhou and Sun, 2019).

Among the related works for testing learning-based implementations, most metamorphic relations are designed to synthesize follow-up inputs without considering the internal operations. The transformation usually includes simple affine operations and composite ones. Metamorphic relations such as brightness modification, rotation, and scaling are affine operations. Metamorphic relations such as inserting objects or various weathers are composite ones. As the inputs of text localization systems are images, we select some existing transformations such as increasing/decreasing brightness (Tian et al., 2018), channel switch (Dwarakanath et al., 2018) in the stability evaluation. However, as the inputs contain text segments, metamorphic relations such as rotation, shearing, and blurring may decrease the performance of text localization, thus leading to an unfair evaluation of stability. On considering these issues, we propose three additional metamorphic relations in text localization. Similar to inserting objects for testing object detection systems (Wang and Su, 2020), one of our metamorphic relations inserts text segments w.r.t. the localized results from original inputs to synthesize follow-ups. The inserted objects in the work are similar to the existing ones in the image. However, in our method, the inserted texts are randomly generated, and the color is different from the background. Different from linear scaling over test features (Dwarakanath et al., 2018) or affine transformation with a 2 × 3 transformation matrix (Tian et al., 2018), our perspective transformation only considers four corners of the image and is executed with a 3 × 3 matrix. Consequently, the modification in perspective transformation is relatively small. Meanwhile, the modified shapes of texts in follow-ups also preserve transformation relations with those in the source.

7. Conclusion

In this paper, we have proposed a methodology to evaluate the stability of text localization systems with metamorphic testing techniques. We have introduced six metamorphic relations w.r.t. the properties of text localization systems and the feature of their inputs. Follow-ups can be generated automatically and compared with the source according to the defined metamorphic relations. The extensive experimentation on both academic and commercial text localization tools reveals many inconsistent outputs, and demonstrates that the methodology is effective in shooting both the advantages and disadvantages of such systems, and evaluating system stability. We have also investigated various image evaluation metrics to analyze the relation between the features of the images and the stability of tools, which can be applied as heuristics for further diagnosis and improvement.

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.

Acknowledgments

This work has been partly funded by Key Research Program of Frontier Sciences, CAS, China, under Grant No. QYZDJ-SSW-JSC036, the CAS-INRIA major project, China under No. 171311KYS B20170027, and the Guangdong Science and Technology Department, China under Grant No. 2018B010107004.

References

Agarwal, R., Srikant, R., et al., 1994. Fast algorithms for mining association rules. In: Proc. of the 20th VLDB Conference, vol. 487, p. 499..

A tensorflow re-implementation of EAST, https://github.com/argman/EAST.

Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson correlation coefficient. In: Noise Reduction in Speech Processing. Springer, pp. 1–4.

Bezryadin, S., Bourov, P., Ilinih, D., 2007. Brightness calculation in digital image processing. In: International Symposium on Technologies for Digital Photo Fulfillment. Society for Imaging Science and Technology, pp. 10–15.

Chatterjee, S., Hadi, A.S., 2015. Regression Analysis by Example. John Wiley & Sons.

Chen, T.Y., Kuo, F., Liu, H., Poon, P., Towey, D., Tse, T.H., Zhou, Z.Q., 2018. Metamorphic testing: A review of challenges and opportunities. ACM Comput. Surv. 51 (1), 4:1–4:27.

Deng, D., Liu, H., Li, X., Cai, D., 2018. PixelLink: Detecting scene text via instance segmentation. in: AAAI, pp. 6773–6780..

Derczynski, L., 2016. Complementarity, F-score, and NLP evaluation. In: LREC.

Dwarakanath, A., Ahuja, M., Sikand, S., Rao, R.M., Bose, R.P.J.C., Dubash, N., Podder, S., 2018. Identifying implementation bugs in machine learning based image classifiers using metamorphic testing. In: ISSTA, pp. 118–128.

Friedman, M., 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Amer. Stat. Assoc. (American Statistical Association) 32 (200), 675–701.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., Vechev, M.T., 2018. AI2: safety and robustness certification of neural networks with abstract interpretation. In: SP, pp. 3–18.

Huang, X., Kroening, D., Ruan, W., Sharp, J., Sun, Y., Thamo, E., Wu, M., Yi, X., 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Comput. Sci. Rev. 37, 100270.

ICDAR2015, https://iapr.org/archives/icdar2015/index.html%3Fp=254.html.

Lin, H., Yang, P., Zhang, F., Review of scene text detection and recognition. Arch. Comput. Methods Eng.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: CVPR. IEEE Computer Society, pp. 3431–3440.

Murphy, C., Shen, K., Kaiser, G.E., 2009. Using JML runtime assertion checking to automate metamorphic testing in applications without test oracles. In: ICST, pp. 436–445.

Pei, K., Cao, Y., Yang, J., Jana, S., 2017. Deepxplore: Automated whitebox testing of deep learning systems. In: SOSP, pp. 1–18..

Schwalbe, G., Schels, M., 2020. A survey on methods for the safety assurance of machine learning based systems. In: ERTS.

Szeliski, R., 2011. Computer vision - algorithms and applications. In: Texts in Computer Science. Springer.

Tian, Y., Pei, K., Jana, S., Ray, B., 2018. Deeptest: automated testing of deep-neural-network-driven autonomous cars. In: ICSE, pp. 303–314.

Vargha, A., Delaney, H.D., 2000. A critique and improvement of the cl common language effect size statistics of mcgraw and wong. J. Educ. Behav. Stat. 25 (2), 101–132.

Wang, S., Su, Z., 2020. Metamorphic object insertion for testing object detection systems. In: ASE.

Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S., 2019. Shape robust text detection with progressive scale expansion network. In: CVPR, pp. 9336–9345.

Xiao, H., Lu, W., Li, R., Zhong, N., Yeung, Y., Chen, J., Xue, F., Sun, W., 2019. Defocus blur detection based on multiscale SVD fusion in gradient domain. J. Vis. Commun. Image Represent. 59, 52–61.

Xie, X., Ho, J.W.K., Murphy, C., Kaiser, G.E., Xu, B., Chen, T.Y., 2009. Application of metamorphic testing to supervised classifiers. In: QSIC, pp. 135–144..

Zhang, J., Harman, M., Ma, L., Liu, Y., 2020. Machine learning testing: Survey, landscapes and horizons. IEEE Trans. Software Eng. early access.

Zhang, M., Zhang, Y., Zhang, L., Liu, C., Khurshid, S., 2018. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In: ASE, pp. 132–142.

Zhou, Z.Q., Sun, L., 2019. Metamorphic testing of driverless cars. Commun. ACM 62 (3), 61–67.

Zhou, Z., Xiang, S., Chen, T.Y., 2016. Metamorphic testing for software quality assessment: A study of search engines. IEEE Trans. Softw. Eng. 42 (3), 264–284.

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J., 2017. EAST: an efficient and accurate scene text detector. In: CVPR, pp. 2642–2651.