What is a good evaluation protocol for text localization systems? Concerns, arguments, comparisons and solutions

Stefania Calarasanu[a,*], Jonathan Fabrizio[a], Severine Dubuisson[b]

[a]*EPITA Research and Development Laboratory (LRDE)*
14-16, rue Voltaire, F-94276, Le Kremlin Bicêtre, France
[b]*Sorbonne Universités, UPMC Univ Paris 06 CNRS,*
UMR 7222, ISIR F-75005, Paris, France

Abstract

A trustworthy protocole is essential to evaluate a text detection algorithm in order to, first measure its efficiency and adjust its parameters and, second to compare its performances with those of other algorithms. However, current protocols do not give precise enough evaluations because they use coarse evaluation metrics, and deal with inconsistent matchings between the output of detection algorithms and the ground truth, both often limited to rectangular shapes. In this paper, we propose a new evaluation protocol, named EvaLTex, that solves some of the current problems associated with classical metrics and matching strategies. Our system deals with different kinds of annotations and detection shapes. It also considers different kinds of granularity between detections and ground truth objects and hence provides more realistic and accurate evaluation measures. We use this protocol to evaluate text detection algorithms and highlight some key examples that

*Corresponding author
Email address: calarasanu@lrde.epita.fr (Stefania Calarasanu)

show that the provided scores are more relevant than those of currently used evaluation protocols.
Keywords: Evaluation protocol, Text detection

1. Introduction

Text detection is an important task in image processing, and many algorithms have been proposed since the last two decades [1]. Hence, text detection systems require a reliable evaluation scheme that provides a ground truth (GT) as precise as possible and a protocol that can evaluate the precision and the accuracy of a text detector with regards to this GT. A solid evaluation protocol should also be able to fairly compare different algorithms. A text detection algorithm can be evaluated differently depending on its output, that can be either boxes surrounding the detected texts, or masks of detected texts after their binarization. One can also directly evaluate the output of an O.C.R.: in such case, the detection algorithm integrates a recognition module and provides as output the text transcription, which is then compared to the true text.

While the output provided by the O.C.R. seems to be the ultimate way to evaluate text detection algorithms, the computed scores do not always correctly reflect the detection accuracy: the transcription can fail because of the distortions of the detected text or its fonts. Furthermore, text transcription is not always necessary, especially in applications for which only the text detection is needed (such as text enhancement or license plate blurring). The evaluation of a text mask is a difficult task as well, mainly because it requires the *true* binarization of the text, that can vary depending on the

text properties (stroke thickness for example). Here again, the evaluation does not focus on the detection results but evaluates both the detection and the binarization (in practice, this binarization is also not necessarily needed).

The simplest and most common way to evaluate a text detection algorithm is then to compare its detection bounding boxes to those that have been manually annotated (i.e. from the GT). This is the common strategy used in most text detection challenges (ImageEval, ICDAR) to evaluate and compare algorithms. However, we have noticed that these evaluation protocols are not reliable. This is due, both to the metrics used for the evaluation, and to the GT annotations [2, 3], that can lead to irrelevant evaluation and comparison of text detection algorithms.

An annotation is sometimes subjective, and therefore it can be difficult to choose how text should be annotated [2]. It is yet possible to construct a dataset only composed of images in which there is no ambiguity for the annotation. However, there is still the problem of tilted or curved texts for which a bounding rectangular box is not appropriate because it can contain a lot of non-text areas. It is then important to define rules for labelling and defining the granularity, *i.e.* the minimal text entity to include into a bounding box. Different levels of granularity can be defined for the GT annotation, depending on the text to detect: the *line*, *word* and *character* levels. The line level is not well suited for tilted text. The character level provides a tedious annotation and promotes connected component approaches. The best granularity level seems to be the word level, even if it is still not the best choice for multi-oriented text.

Choosing good metrics to compare detections that do not correctly match

the GT objects is also a complex task. Most of the metrics can not efficiently deal with the difference of granularity levels between the GT and the detections. For example, if the GT is at word level and the detection at line level, the score will be most of the time over-penalized. Moreover, as pointed by Wolf and Jolion in [4], a single metric cannot truly describe the complex behavior of a localization algorithm, namely separating the quantity nature ("*how many GT boxes were detected*") from the quality aspect ("*how well the GT boxes were detected*") of a detection. Although these issues were addressed in the literature (see Section 2), the proposed solutions are still not satisfactory.

Because of all these limitations, researchers do not have any robust tool to get a representative evaluation of their algorithm and a fair comparison with other algorithms. For example, the authors in [5] claim their scores are too low because the ICDAR2013 protocol does not correctly evaluate line level detections. Hence, some other works that provide detections at line level [6, 7] have proposed to change the GT annotation of ICDAR2005 dataset from word to line level to be less penalized. However, this does not permit a correct comparison with other scores obtained using the same database with the word level annotation. Sun *et al.* [8] manually split their line level detections in order to use the ICDAR2013 protocol and compare their results. Manual splitting is also a problem because it makes irrelevant the comparison with other detectors integrating an automatic splitting step (or ever no splitting). Du *et al.* [9] have also split their line level detections into words, however, no detail about the splitting procedure is given. Due to the lack of a fair evaluation protocol, many works [10, 11] evaluate their

algorithm by using others protocols. However, this gives an inconsistent comparaison to other algorithms.

Only few interest has been given to the evaluation protocol of text detection algorithms. Some works [12, 13, 14] do not mention at all what protocols are used for the evaluation, while others [15, 16, 17, 18, 19, 20, 21, 22] limited their explanations to "*standard recall, precision and F-Score*" without any further details concerning their computation or matching strategies. DetEval is probably the most frequently used evaluation protocol. Its framework is tunable and hence its configuration should always be specified when used. However, only few works [23, 24] specify the used parameters, while many do not mention them [25, 26, 27, 28, 29, 30]. All these examples prove a need of revising the current evaluation protocols.

In this article, we propose a new evaluation protocol providing many advantages compared to the most common used, listed below.

- It can handle different detection granularities. For that, we propose a two-level rectangular GT annotation, which allows an equitable comparison between algorithms having different granularity outputs.

- It provides a clear identification of the matching strategy between a GT object and a detection (one-to-one, one-to-many, many-to-one and many-to-many cases) and adapts the two quality metrics (coverage and accuracy) to each type of matching.

- It computes both quantity and quality recall and precision scores to give a full comprehension of a detector's behavior.

- It can be easily adapted to manage any irregular text representation,

such as polygonal, elliptic or free-form ones.

This article is organized as follows. Section 2 first gives a short survey of the existing metrics and evaluation protocols for text detection algorithm evaluation and comparison. Section 3 presents our evaluation procedure called EvaLTex. We first define our two-level annotation that permits to deal with different detector's output granularities (Section 3.1). Then we detail our matching procedures to avoid over or under penalizations while matching detections and ground truth objects (Section 3.2). We also propose a generalization of our protocol to evaluate a set of images and derive quality and quantity scores for the detection (Section 3.3). Finally, we show how EvaLTex can also manage free form annotations (Section 3.4). Section 4 is dedicated to the validation of our evaluation framework in the context of text detection and its comparison to other evaluation protocols. In particular, we show that the currently used evaluation protocols can not efficiently manage many detection scenarios and that our method provides more logical scores. Finally, concluding remarks and perspectives are given in Section 5.

2. Evaluation protocols: related works

In the past decades, various datasets and performance measures have been proposed for text localization tasks and some of them are listed in Table 1.

The performance measures that have been mainly used to compare and evaluate text detectors are the recall scores (*i.e.* number of correctly detected texts divided by the total number of GT objects) and precision scores (*i.e.* number of correctly detected texts divided by the total number of detected texts). If an algorithm detects too many text regions, its precision rate

Table 1: Datasets used for text localization tasks. The symbol "−" suggests that no information is available. In the "Evaluation protocol" column, "$\surd$" and "X" symbols denote the existence or not (respectively) of an evaluation protocol associated to the corresponding dataset. In the "Annotation" column "C", "W" and "L" correspond to the level of granularity (character, word and line levels respectively), while "BB" refers to a bounding box representation.

Dataset	**Text category**	**Evaluation protocol**	**Annotation**	**Size**	**Characteristics**
ICDAR′03/05 [31, 32]	Scene	$\surd$	W/BB	509 images	Horizontal, English text
MSRA-I [33]	Graphic/ Scene	$\surd$	W/BB	45 images	Horizontal, English Spanish and Chinese text
Sign Evaluation Data [34]	Scene	X	C/BB	95 text regions	English text
Microsoft [35]	Scene	X	−	307 images	Repeating patterns Background with vegetation
OCST [36]	Scene	X	W/BB	−	Difficult text Incomplete annotation
KAIST [37]	Scene	X	C,W/BB		English and Korean text Various lighting condition
SVT [38]	Scene	X	W/BB	350 images	Frontal, horizontal, English text; Incomplete word annotation
SVT-Perspective [39]					Perspective text
NEOCR [40]	Scene	X	W/straight& oriented BB	659 images	8 different languages
OSTD [41]	Scene	X		80 images	Multi-oriented text
ICDAR′11 [42]	Digital/Scene	$\surd$	W/BB	552/484 images	Horizontal, English text
MSRA-TD500 [43]	Scene	X	L/BB	700 images	Multi-oriented English/Chinese text
MSRA-TD500 Word [39]			W/BB		
ICDAR′13 [44]	Digital/Scene	$\surd$	W	551/462	Horizontal English text
MASTER [24]	Scene	$\surd$	W/BB	134 images	Various text scripts
CUTE80 [45]	Scene	$\surd$	L/polygon points	80 images	Curved, perspective text Complex backgrounds
HUST-TR400 [13]	Scene	X	W/BB	400 images	English texts Arabic numbers

will decrease, while if it detects too few texts, its recall rate will decrease. Performing a fair evaluation requires to determine if a detection is correct and to correctly match it with its corresponding GT object (particularly if their granularity is different).

2.1. Correct detections

Most of the current evaluation protocols consider a detection as correct if the overlap area between its region and the corresponding GT object is sufficiently large [46, 44, 47, 48, 4]. This gives a binary evaluation, whether this minimum overlap constraint is satisfied or not. Hence, if we compare the two detections of Figure 1, one that partially covers a text (without satisfying the overlap constraint) and one that entirely misses it, both will unfairly get the same score. Despite its irrelevant scoring, this approach was however used during the latest ICDAR competitions [32, 31, 49, 42].

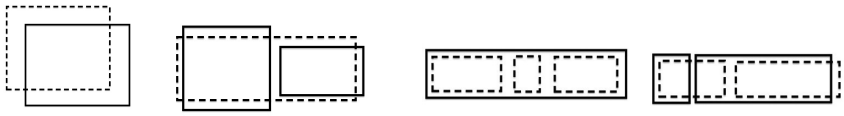
(a) Text not detected; 171 ICDAR 2013 Inkam method

(b) Partial text detection (red rectangle); 148 ICDAR 2013 Text_detector_CASIA

Figure 1: An example of irrelevant score. Both methods got recall and precision scores equal to 0 during the ICDAR2013 competition evaluation protocol.

2.2. Matching strategies

Another challenge concerns the matching of detections and GT objects, particularly if their granularity is different. The matching consists in establishing the links between the detection and the GT objects. Four types of matchings can be considered, as it can be seen in Figure 2: (a) one-to-one: one detection matches exactly one GT object ; (b) one-to-many: multiple detections match one GT object ; (c) many-to-one: one detection matches multiple GT objects ; (d) many-to-many: mix of cases (b) and (c).



(a) One-to-one (b) One-to-many (c) Many-to-one (d) Many-to-many

Figure 2: Matching cases (GT is represented by dashed rectangles and detections by plain line rectangles).

2.3. Existing protocols

Ma *et al.* [50] proposed a word level evaluation, where GT objects are clustered with respect to a proximity criterion. Hence, single-row and multi-row merges are equally allowed, but the precision is still penalized when such cases occur. Also, if a GT text box is detected several times, only the maximum overlap area is considered, which is a severe penalization for algorithms that can correctly detect a text through two (or more) detections. The evaluation framework in [46] uses a multi-level text annotation (pixel, atom, word and line) but can not handle word and line level texts represented with bounding

boxes. In the evaluation protocol introduced by Hua *et al.* [33], the GT annotation is done at the line level and then penalizes methods whose detection outputs are at word or character level. The method proposed by Nascimento *et al.* [48] can evaluate the percentage of different types of matchings, including splits and merges by generating multiple GT interpretations. However, no global measurements are proposed, which makes the comparison between different algorithms difficult to interpret. This problem also occurs in the protocol presented by Mariano *et al.* [47], who proposed a set of 7 evaluation metrics but not a global one. The VACE metric described in [51] consists in an overall performance measurement, Frame Detection Accuracy (FDA), between all GT objects and detections. Nevertheless, this metric does not provide a clear separation between recall and precision. The CLEAR metrics, also proposed in [51], compute the accuracy and the precision of a text detector separately based on the true coverage area between the GT and the detections but the authors do not explain the evaluation of different matching scenarios. Anthimopoulos *et al.* suggested in [52] an evaluation method based on the number of detected characters estimated as their width/height ratio. The MSRA-TD500 [43] framework is able to handle oriented texts. The protocol considers a detection as correct if the angle between it and its corresponding GT object as well as their overlap ratio satisfy two thresholds. If multiple detections match the same text line, they are considered as false positives. The evaluation protocol associated to CUTE80 dataset consists in establishing the minimum intersection area between the GT and the detection polygon points of a curved text line. However, all matching types are treated equally. In [41], the authors proposed an evaluation protocol

that can deal with inclined text line but only computes a precision value. In [53, 54, 55, 56] the authors proposed an evaluation framework at block level that does not penalize partial text lines.

Wolf *et al.* [4] proposed a more complex text detection evaluation scheme, named DetEval, based on performance graphs, which considers the precision and recall rates as quality scores. It can manage the one-to-many and many-to-one cases, but uses parameter functions to penalize both cases. The ZoneMap metric [57] is a generalization of [58] and [4] that computes different error rates based on the overlapping areas in tables.

The ICDAR [44] Robust Reading Competition is considered as the main reference for text detection and localization algorithm comparisons. The evaluation method used during this competition is based on the algorithm proposed in [4], assumed to be the most efficient one. But this protocol, as it is used during the ICDAR competition, faces many problems. First, the overlapping area ratio constraint misclassifies many GT text boxes during the matching protocol. This results in low scores, even when the detected boxes substantially overlap the GT ones. Second, the scattering scenarios are poorly treated. Finally, the annotation is done at the word level which frequently assigns high penalties to text line detections, as seen in the results published in [44].

In the next section, we propose a new evaluation procedure, which solves most of the previously mentioned problems.

3. Our evaluation protocol: EvaLTex

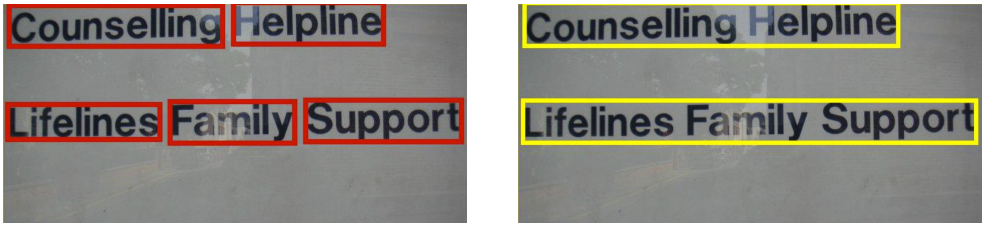
3.1. Rectangular ground truth annotation

Annotating the GT for text detection is not an obvious task and relies on the target application as well as on the rules chosen for this annotation. We indeed have to decide the minimum text size a detector should be able to deal with, or also if a word such as "COCA-COLA" should be annotated as a single GT object or as two separate ones. While some of these issues remain debatable, others, such as the granularity difference between the GT and detections can be easily overcome, as it will be shown in Sections 3.2.3 and 3.2.4. Many evaluation protocols that do not deal with different granularities can sometimes severely penalize one algorithm but not another, while these should be scored equally. A solution is to deal with multiple GT annotation levels.

In our approach, we introduce a two-level annotation. Each GT object is first annotated at the world level using a single rectangular box. Then, GT word boxes are manually grouped into regions. Here, we consider word text boxes as part of a same region if they are horizontally (resp. vertically) aligned and have similar heights (resp. widths), but different region grouping could also be considered as long as the text area within one region is larger than the non text area (see Figure 3 for examples of incorrect grouping). A region is therefore a rectangular box containing several text objects annotated at word level. Figure 4 shows an example of the proposed two-level annotation of the GT.

We have chosen this two-level annotation for two main reasons. First, we do not want to reject a detection when its area covers more than one GT word

Figure 3: Examples of invalid text region annotations (black rectangles): the non textual area within these regions is larger than the text area.



(a) (b)

Figure 4: Two examples of ground truth annotation: (a) at word-level; (b) at region-level.

text box (case of many-to-one detections) if these ones belong to the same GT region. In such case, the precision score will then not be penalized. We also want to provide a comparable and equivalent evaluation of algorithms whose outputs are similar, but at different granularity levels (*i.e.* word and line-level). The proposed solution is detailed in Section 3.2.4.

3.2. Matching strategies and performance measurements

Let $G = (G_1, G_2, ..., G_m)$ be the set of GT text boxes (tags) and $D = (D_1, D_2, ..., D_n)$ the set of the detections, with m (resp. n) the number of

objects in G (resp. in D). Then, for each G_i matched to detection D_j, we define Cov_i as their coverage area and Acc_i as the detection accuracy by:

$$Cov_i = \frac{Area(G_i \bigcap D_j)}{Area(G_i)} \tag{1}$$

$$Acc_i = \frac{Area(G_i \bigcap D_j)}{Area(D_j)}. \tag{2}$$

We also assign a value $match_{G_i}$ (resp. $match_{D_j}$) to each G_i (resp. D_j), defined by:

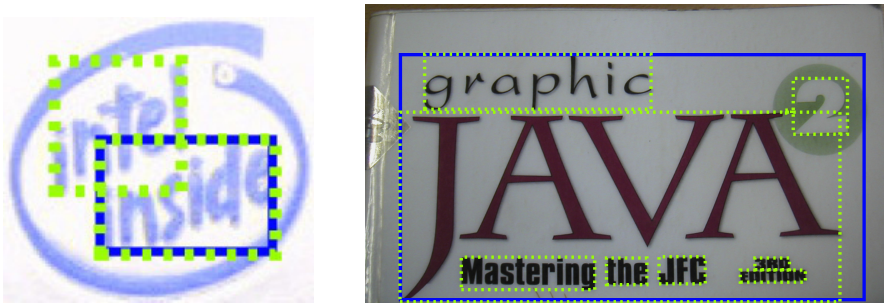
$$match_{G_i} = \begin{cases} 1 & \text{if } \exists\, D_j \mid Area(G_i \bigcap D_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$match_{D_j} = \begin{cases} 1 & \text{if } \exists\, G_i \mid Area(G_i \bigcap D_j) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

3.2.1. Filtering procedure

Before identifying the type of a match (in our case we consider the 4 possible types, see Section 2.2), a filtering procedure is first used to determine, when a detection covers several GT objects, to which of them it can be matched. Figure 5a illustrates a case of two overlapping GT objects (in dashed green) because of the tilted text. The word "inside" should be matched to the blue detection, while "intel" should not. Hence, when a many-to-one match occurs, we first determine if a detection D intersects more than one GT object. If so, we match detection D to GT object G and not to G' (we can generalize this reasoning to as many GT objects as needed) if the following area constraint is satisfied:

$$Area(G' \bigcap D) - Area(G \bigcap G') \leq t \cdot Area(G'), \tag{5}$$



(a) (b)

Figure 5: Filtering procedure: matching detected objects (blue) with GT boxes (dashed green); (a) the tilted text causes an overlap of GT text boxes, (b) the character height variation (see the letter "J") causes the inclusions of GT text boxes.

where t is a threshold that regulates the overlap area rate. In our experiments, t was set to 0.1, to filter GT objects with a small overlap area. By increasing t, we could reject valid GT objects that are part of a many-to-one matching. Figure 5b illustrates the case of text inclusion: the GT object for word "JAVA" includes other GT objects. In this case, we should match the detection to words "JAVA", "graphic" and "2", while the other words should have been detected separately in order to be correctly matched.

During the next stage, coverage and accuracy scores are computed differently for each GT object depending on the matching strategy, as described in the following sections.

3.2.2. One-to-one match

The detection is evaluated using the coverage (Eq. (1)) and accuracy (Eq. (2)) scores. Assuming that we never get a perfect match, we use a margin error m_e to extend or reduce the area of GT object G_i, computed as

follows:

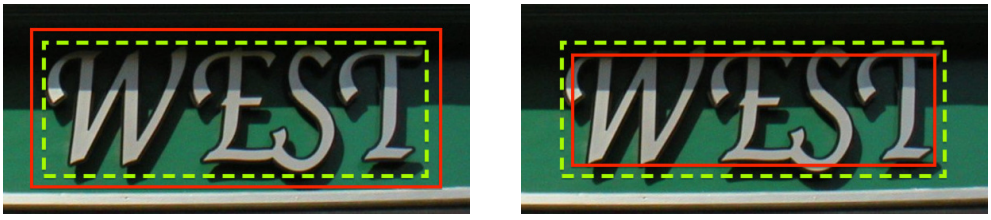
$$m_e = \begin{cases} t_m \cdot \frac{Area(G_i)}{height(G_i)} & \text{if } height(G_i) \geq width(G_i) \\ t_m \cdot \frac{Area(G_i)}{width(G_i)} & \text{otherwise} \end{cases} \tag{6}$$

where t_m is a parameter that controls the thickness of the margin error.

Let $[x_{G_i}, y_{G_i}, w_{G_i}, h_{G_i}]$ define the GT text box G_i, with x_{G_i} and y_{G_i} its left upper corner coordinates, and w_{G_i} and h_{G_i} its width and height respectively. Let Ge_i and Gr_i be the extended and the reduced text boxes (see example in Figure 6) of G_i, with:

$$Ge_i : [x_{G_i} - m_e, y_{G_i} - m_e, w_{G_i} + 2 \cdot m_e, h_{G_i} + 2 \cdot m_e] \tag{7}$$

$$Gr_i : [x_{G_i} + m_e, y_{G_i} + m_e, w_{G_i} - 2 \cdot m_e, h_{G_i} - 2 \cdot m_e] \tag{8}$$



(a) enlarged text box (b) reduced text box

Figure 6: Illustration of the extended and reduced boxes (red), obtained from a GT box (dashed green).

For any one-to-one match between a detected box D_j and a GT box G_i, the accuracy is computed by considering the enlarged GT text box Ge_i:

$$Acc_i = \frac{Area(Ge_i \bigcap D_j)}{Area(D_j)}, \tag{9}$$

while the coverage is computed using the reduced GT text box Gr_i:

$$Cov_i = \frac{Area(Gr_i \bigcap D_j)}{Area(Gr_i)} \tag{10}$$

Consequently, the higher t_m the higher the coverage and accuracy values. Hence, it is not recommended to give a very high value to this parameter as it might degrade the detection evaluation. In our experiments, t_m is set to 0.1, a reasonable value that allows small imprecisions for detections.

3.2.3. One-to-many match

The one-to-many case is illustrated in Figure 7 where the word "Yarmouth" is matched to 2 different detection boxes.

Figure 7: One-to-many case for "Yarmouth" word: one object in G (dashed green) is matched to multiple boxes in D (blue).

This scenario implies a fragmentation level given by the number of detections (s_i) associated to one GT object G_i. We use the fragmentation to penalize the coverage of G_i in the following manner:

$$Cov_i = Cov_i^u \cdot F_i \tag{11}$$

where Cov_i^u represents the union of all intersection areas between Gr_i and all detections D_j, $j \in [1, s_i]$, normalized by the area of Gr_i, defined as:

$$Cov_i^u = \frac{\bigcup_{j=1}^{s_i} Area(Gr_i \bigcap D_j)}{Area(Gr_i)}; \tag{12}$$

F_i represents the fragmentation index suggested by Mariano *et al.* [47]:

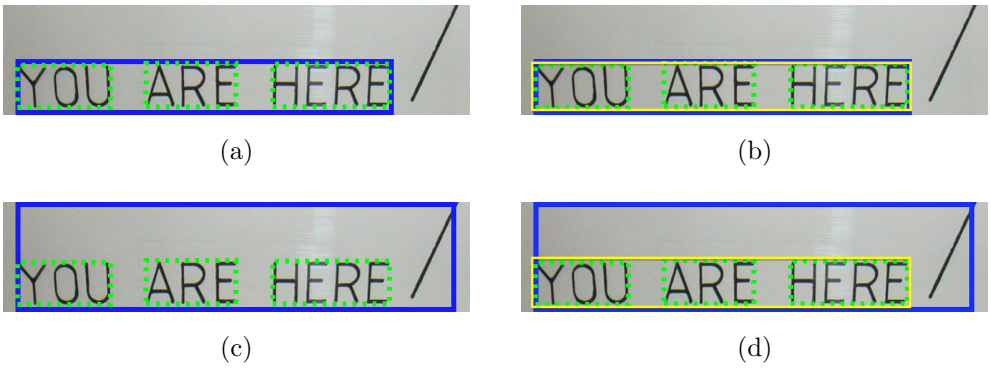
$$F_i = \frac{1}{1 + \ln(s_i)} \tag{13}$$

Similarly, the corresponding accuracy G_i is defined as the union of all intersection areas between Ge_i and detections D_j, $j \in [1, s_i]$, normalized by the total of detection areas:

$$Acc_i = \frac{\bigcup_{j=1}^{s_i} Area(Ge_i \bigcap D_j)}{\bigcup_{j=1}^{s_i} Area(D_j).} \tag{14}$$

3.2.4. Many-to-one match

The many-to-one case implies that several GT objects correspond to one single detection. This case is illustrated in Figure 8. Our protocol considers



(a) (b) (c) (d)

Figure 8: Many-to-one case: a detection (blue) matches several GT objects (dashed green); (a) a detection box close to the GT objects, (b) a detection box close to the GT objects grouped into a region (yellow), (c) a coarser detection of the GT objects, (d) a coarser detection of the GT objects grouped into a region (yellow).

a many-to-one match as several one-to-one cases. Hence, the coverage for each GT objects G_i, $i \in [1, m_j]$, with m_j the merge level of the detection box

D_j, is:

$$Cov_i = \frac{Area(Gr_i \bigcap D_j)}{Area(Gr_i)} \tag{15}$$

While the coverage only considers the amount of valid matched GT objects, the detection accuracy takes into account the quantity of non textual areas (areas outside the GT) that have been detected. Consequently, if a detection matches several GT objects, the non textual area coming from the inter-object spacing contributes to the penalization of the accuracy score. Then, a fair comparison between a word level detection and, for example, a line level detection is not possible. Hence, many one-to-one detections would always outweigh one many-to-one detection. However, in some cases, word level and sentence level detections should be treated equally. Our two-level GT annotation solves this problem and provides a better comparison between different detection granularities. We then assume that the area of a text region does not contain any non textual area and now consider the spacing area between GT objects belonging to a same region as valid text areas.

Our protocol computes both the coverage and accuracy for each GT object, while traditional approaches assign the coverage to GT objects and accuracy to detections. To compute the accuracy for each G_i, we first match it to a detection. Therefore, the detection area is split to be matched to its corresponding m_j GT objects. Figure 8c shows a many-to-one case, with three GT boxes and one detection whose area is larger than the text one. We define $TextArea_{D_j}$ as the union of all GT text areas covered by the detection box, and $nonTextArea_{D_j}$ the rest of the detection area, *i.e*:

$$TextArea_{D_j} = Area(\bigcup_{i=1}^{m_j}(Ge_i \bigcap D_j)) \tag{16}$$

$$nonTextArea_{D_j} = Area(D_j) - TextArea_{D_j}. \tag{17}$$

The accuracy associated to each matched G_i is:

$$Acc_i = \frac{Area(Ge_i \bigcap D_j)}{Area(D_{j,i})}, \tag{18}$$

where $Area(D_{j,i})$ is the detection area covering each extended box Ge_i, defined as:

$$Area(D_{j,i}) = \frac{Area(Ge_i)}{TextArea_{D_j}} \cdot nonTextArea_{D_j} \tag{19}$$

We now define a text region Reg, as the box bounding a set of GT objects (see the yellow boxes in Figures 8b and 8d). Then, we redefine the $TextArea_{D_j}$ as the union of all text regions Reg_k within the detection box:

$$TextArea_{D_j} = Area(\bigcup_{k=1}^{r_j}(Reg_k \bigcap D_j)), \tag{20}$$

where r_j represents the number of GT regions covered by D_j.

3.2.5. Many-to-many match

Figure 9: Many-to-many case: a mix of one-to-many and many-to-one cases.

The many-to-many occurs when the same GT objects are involved simultaneously in a one-to-many and a many-to-one match. This is illustrated

in Figure 9. There is a many-to-one match because a detection (in blue) includes the word "HEALTHY" and a part of the word "COLCHESTER". The one-to-many match is due to the word "COLCHESTER" that is covered by 2 detections.

In our approach, the many-to-many match is treated as a one-to-many followed by a many-to-one match. Therefore, the coverage and accuracy are computed using the equations defined for the many-to-one case (Section 3.2.4) and for the one-to-many case (Section 3.2.3). For example, the word "HEALTHY" is part of a many-to-one scenario: its coverage is computed using the Equation (15) and its accuracy using the Equation (18). The coverage and accuracy of word "COLCHESTER", involved in a one-to-many match, are computed using Equations (11) and (14) respectively.

3.3. Evaluation protocol on a set of images

The scores presented in the previous sections evaluate the *quality* nature of a detection: how well an individual GT text box has been detected and the precision of each valid detection. However, when dealing with a whole dataset (*i.e.* a set of images), it is also necessary to evaluate the *quantity* nature of the detections, namely how many GT objects or false positives were detected on the whole database. The distinction between the quantity and quality aspects of a detection is useful for a better comprehension of the detection results. As pointed in [4], *"a recall value equal to 50% can mean that either 50% of the GT text boxes were matched at a 100% rate or that 100% of the GT boxes were detected at a 50% rate. Similarly, a 50% precision result can mean that the total GT area covered by the detection boxes represents 50% of the total detection areas, but it can also mean that*

only 50% of the detection boxes correctly cover the GT, while the other 50% are false positives". Consequently, the quality values measure the matching area rates between the GT and detection boxes, whereas the quantity values represent the amount of valid matchings between the GT and detections as well as the amount of false positive.

Then we compute, for the whole set of images, both quality and quantity overall recall and precision scores and combine them to get two global scores. Let tp be the number of true positives (*i.e.* of matched objects in G) and fp the number of false positives (*i.e.* of objects in D that have no correspondence in G), *i.e*:

$$tp = \sum_{i=1}^{m} (match_{G_i} = 1), \tag{21}$$

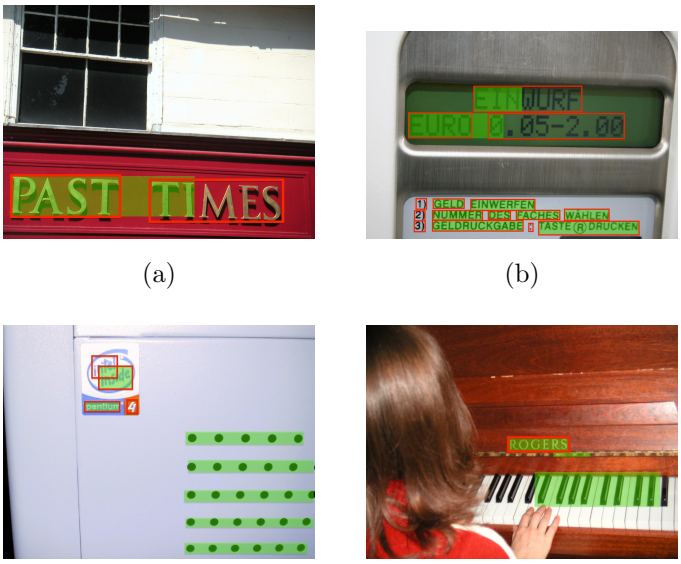
$$fp = \sum_{j=1}^{n} (match_{D_j} = 0) \tag{22}$$

Here m and n are respectively the number of GT objects and detections over the whole dataset.

Table 2: Quantity, quality and global scores for each image, as well as for the whole set of images in Figure 10.

Fig.	**m/tp/fp**	R_{quant}	P_{quant}	R_{qual}	P_{qual}	R_G	P_G	F_G
10a	2/2/0	1	1	0.66	0.74	0.66	0.74	0.69
10b	15/11/0	0.73	1	0.86	0.92	0.63	0.92	0.75
10c	4/2/5	0.5	0.28	1	1	0.5	0.28	0.36
10d	1/1/2	1	0.33	1	1	1	0.33	0.5
Set	22/16/7	0.72	0.69	0.86	0.91	0.63	0.64	0.63

For a a many-to-one case (Section 3.2.4), we split the detection into several areas, each one covering a GT object. For the one-to-many case (Sec-



(a) (b)

(c) (d)

Figure 10: A set of four images; GT objects are red rectangles and detection green ones.

tion 3.2.3), we merge the corresponding detection areas and compute a single precision score (Equation (14)).

Then, we define two metrics, that measure how many GT objects were detected (quantity recall R_{quant}) and how many detections have a correspondence in the GT (quantity precision P_{quant}), given by:

$$R_{quant} = \frac{tp}{m}, \tag{23}$$

$$P_{quant} = \frac{tp}{tp + fp}, \tag{24}$$

Moreover, we compute a quality recall, R_{qual}, measuring the overlap area for all valid matchings between the detections and the GT objects and a quality

precision, P_{qual}, estimating the total detection accuracy, defined as:

$$R_{qual} = \frac{\sum_{i=1}^{m} Cov_i}{tp} \tag{25}$$

$$P_{qual} = \frac{\sum_{i=1}^{m} Acc_i}{tp}, \tag{26}$$

Since a good set of metrics should reflect both the quantity and the quality nature of a detection, we propose the two following global recall and precision scores:

$$R_G = \frac{\sum_{i=1}^{m} Cov_i}{m}, \tag{27}$$

$$P_G = \frac{\sum_{i=1}^{m} Acc_i}{tp + fp}, \tag{28}$$

The quality components of these global metrics are given by the numerator values $\sum_{i=1}^{m} Cov_i$ and $\sum_{i=1}^{m} Acc_i$ (sum of all GT object qualities Cov_i and Acc_i). The quantity nature is given by the mean of the coverage quality components of the m GT objects, and the accuracy quality components over the total number of detections $tp + fp$. Indeed, R_G and P_G are equivalent to the product of the quality and the quantity components:

$$R_G = \frac{\sum_{i=1}^{m} Cov_i}{m} = R_{quant} \cdot R_{qual}, \tag{29}$$

$$P_G = \frac{\sum_{i=1}^{m} Acc_i}{tp + fp} = P_{quant} \cdot P_{qual}, \tag{30}$$

The F-Score F_G is used to measure the overall performance of a detection

algorithm and is defined as the harmonic mean of the global recall and precision values:

$$F_G = \frac{2 \cdot R_G \cdot P_G}{R_G + P_G} \tag{31}$$

Figure 10 shows an example for a set of four images and their corresponding GT and detections. The evaluation for each image and for the whole set using our proposed metrics is summarized in Table 2.

3.4. Extension to any text representation

A rectangular representation of texts can generate errors during the matching process, in cases of inclined, curved of circular texts, as it can be seen in Figure 11. Although we proposed a procedure in Section 3.2.1 to discard "unlikely" matched GT objects, this cannot ensure that all matchings will be correct (see Section 4.2). For texts that are neither horizontal, nor vertical, typically texts that are encountered in urban scenes, a representation using a free-form mask is more adapted.

In this section we show how to extend our EvaLTex protocol (matching strategies and performance metrics) to any irregular text representations (also called *masks*), such as polygonal, elliptic or even free-form shapes. Using a mask representation implies the following changes:

- text objects are represented by irregular masks;
- the extension and reduction of GT object regions (Equations (7) and (8)) are computed using dilation and erosion morphological operations on text masks;
- we consider only one level of annotation (word or region) but still manage different granularities.

(a) with rectangles (b) with masks

Figure 11: Different shapes for GT annotation (red). (a) The rectangular GT box bounding word "Pago" encloses the ones of "1888" and highly overlaps the ones of "SINCE", "PREMIUM" and "FRUIT". (b) We avoid such problems by using mask annotations.

4. Experimental results and discussions

In this section, we show the efficiency of our proposed method when using a rectangular representation (see Section 4.1) or a mask representation (see Section 4.2).

4.1. Experimental results using the rectangular representation

We evaluate the detection results on the *Challenge 2* dataset used during the ICDAR 2013 Robust Reading competition [44]. This dataset contains 233 images of natural scene texts and an associated GT annotated at the word level. We use the same word level annotation, but also our region labels (Section 3.1), to introduce another level of granularity.

To illustrate the advantages of EvaLTex we compare it to three commonly used evaluation protocols in the text detection field (ICDAR2003, ICDAR2013 and DetEval). A detailed comparison of matching strategies and detection scores is given in Section 4.1.1, and the interest of using our two-level annotation in Section 4.1.2.

4.1.1. Comparison to other evaluation methods

ICDAR2013 ***evaluation protocol***. We compare our evaluation protocol with the one used during the ICDAR 2013 Robust reading competition [44] for two reasons: (i) it is up-to-date and represents what is commonly done and admitted in text detection evaluation, and (ii) all results are publicly available, making the comparison easy.

The ICDAR2013 protocol relies on the evaluation framework introduced in [4]. It uses the proposed area precision and recall thresholds, which are set to 0.8 and 0.4 respectively, and which control the matching between GT objects and detections. Moreover, a lower weight is assigned to one-to-many matches, since the output is at the word level, while text-line level detections (many-to-one matches) are supposed not to be penalized [44]. However, we will show that this is not always true, and that many scores are erroneous.

Next, we give some scores provided by our matching algorithm (Figure 12), on the detector TextDetection [2] that participated to ICDAR 2013 challenges. We compare our matching method with the one of ICDAR2013. The corresponding scores are given Table 3.

Figures 12a and 12b illustrate a one-to-one case for which the recall and precision scores are over-estimated by ICDAR metrics. First, although the detection missed the first letter of the word “AUSTRALIA”, the recall rate

Table 3: Scores obtained for cases in Figure 12 using ICDAR and our proposed metrics.

Image	**Method**	**Recall**	**Precision**	***F*-Score**
Fig. 12a	ICDAR2013	1	1	1
	EvaLTex	0.9186	1	0.9575
Fig. 12b	ICDAR 2013	0.5	1	0.6667
	EvaLTex	0.5	0.5919	0.5421
Fig. 12c	ICDAR2013	0.625	0.7143	0.6667
	EvaLTex	0.8102	1	0.8952
Fig. 12d	ICDAR2013	0.90	0.8667	0.883
	EvaLTex	0.7806	1	0.8768
Fig. 12e	ICDAR2013	0.6667	1	0.8
	EvaLTex	1	1	1
Fig. 12f	ICDAR2013	0.3333	1	0.5
	EvaLTex	1	0.6245	0.7688
Fig. 12g	ICDAR2013	0.6667	0.6667	0.6667
	EvaLTex	0.8404	1	0.9132
Fig. 12h	ICDAR2013	0.6	1	0.75
	EvaLTex	0.9032	1	0.9491

is set to 1 (Fig. 12a). Similarly, even if the area of the detection box for the word "moto" is considerably larger than the GT object, its precision rate is 1. The ICDAR2013 approach scores a GT object with a binary recall (1 or 0), depending on whether the overlapping area between GT and detection respects or not a threshold. However, in many cases, this does not provide a fair comparison between algorithms. For example, if an algorithm detects the whole word "AUSTRALIA", it will get the same score as the detection shown in Figure 12a. Conversely, our metrics give a more precise and realistic evaluation because they take into account the real overlap match area, and then provide a better comparison between algorithms.

As shown in Figures 12c and 12d, ICDAR2013 metrics can consider the one-to-many match in different ways. In Figure 12d, the word "POSTPAK" is detected by two boxes, both considered as correct. In Figure 12c we

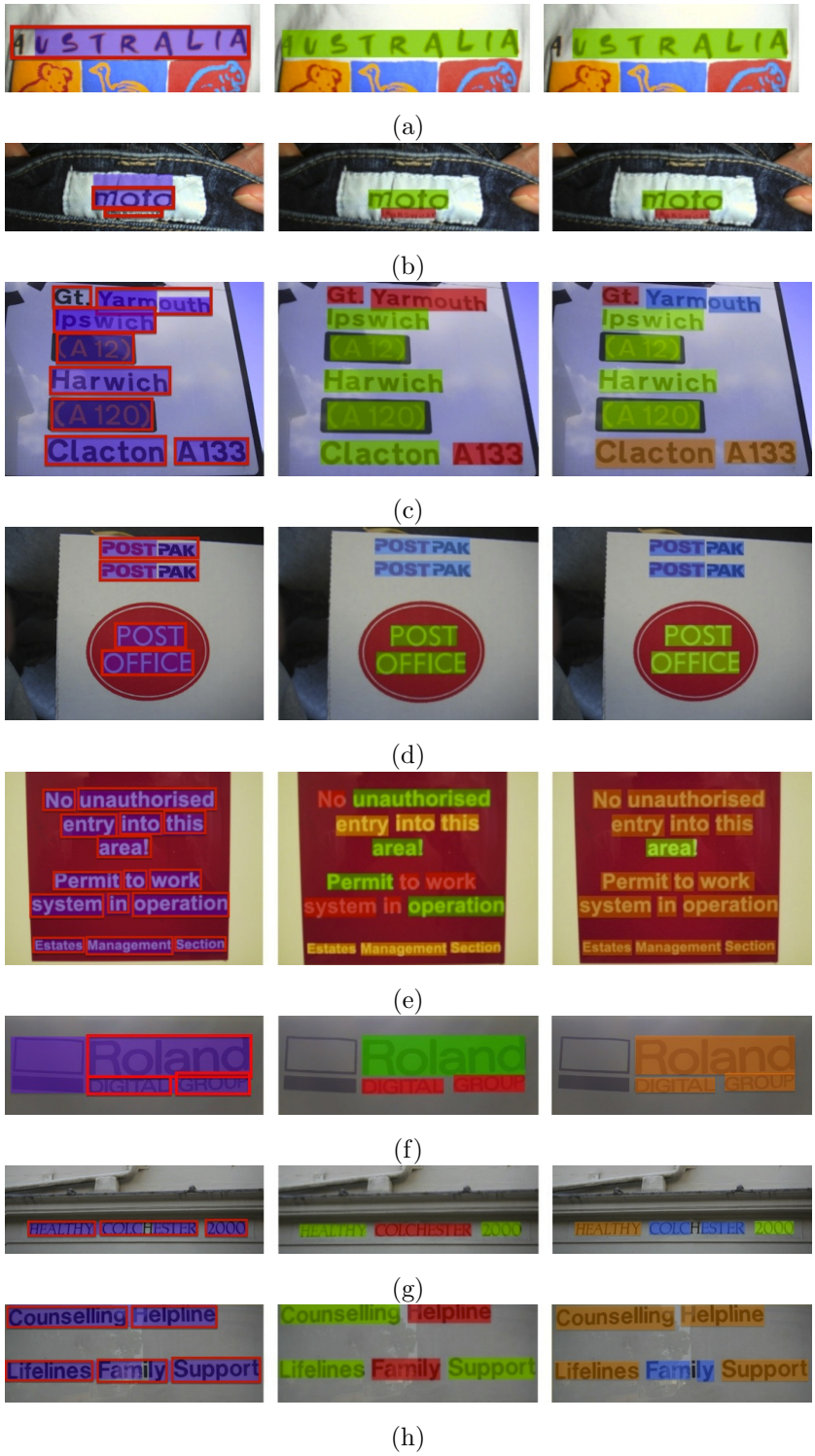


Figure 12: Matching examples; left: GT (red rectangle) and detections (solid purple rectangles); center (ICDAR2013) and right (EvaLTex): mismatched GT objects (solid red rectangles), one-to-one matched GT areas (solid green rectangles), many-to-one matched GT areas (solid yellow rectangles), one-to-many matched GT areas (solid blue rectangles).

have a similar scenario for the word "Yarmouth", but here none of the two detected boxes is considered as valid because in both cases the overlap area is too small. Moreover, the two detections are counted as false positives, which unfairly penalizes the final scores. Firstly, it decreases significantly the precision rate because the two detected boxes are erroneously counted as false positives, and secondly, it decreases the recall rate by not matching the two detected boxes to the GT. On the contrary, our method correctly recognizes the one-to-many cases and matches the two detected boxes in both cases, but punishes the fragmented detection by penalizing the recall score, as seen in Section 3.2.

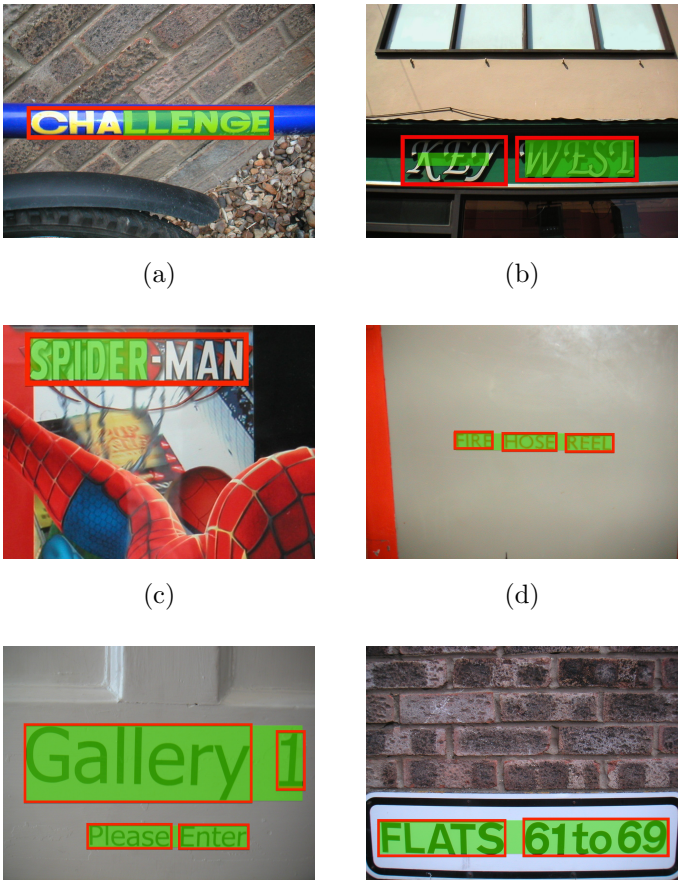
Figures 12e and 12f show a problem of inconsistency during the many-to-one matching. In Figure 12e, the detection is at a line level. Only the second and last lines are correctly matched, while the other detected text lines are associated to the GT text box having the largest area within that line ("unauthorized" in the first line, "Permit" in the third one and "operation" in the fourth one). The unmatched GT text objects ("No", "to", "work", "system", "in") are considered as false positives. ICDAR2013 metrics then over punish the many-to-one cases by frequently considering them as one-to-one. On the contrary, our protocol correctly matches all text lines and leads to a recall equal to 1. We have a similar problem when detections cover multiple text lines (Figure 12f). The word "Roland" is matched by ICDAR2013 protocol, while the two other words are discarded. Hence, their recall is penalized, while their precision is not. Our method considers all words as detected, hence the recall rate is set to 1. Nevertheless, we assign a low precision rate, due to the presence of the logo in the left part of the

detected box.

Finally, the many-to-many case is illustrated in Figures 12g and 12h. The word "COLCHESTER" in Figure 12g corresponds to both a many-to-one and a one-to-many match. Nevertheless, the ICDAR2013 matching protocol rejects it and matches only the word "HEALTHY", whereas our algorithm validates both text boxes, but penalizes the recall due to its split detection. If we look at the second line in Figure 12h we observe that the word "family" is covered by two detections (one-to-many). Both detections involve a many-to-one case, the first one corresponding to the words "Lifelines" and "family", while the second one to the words "family" and "Support". The ICDAR2013 matching algorithm considers GT text boxes as matched those containing the words "Lifelines" and "Support", and classifies the word "Family" as missed. This leads again to an unfair comparison: if another localization algorithm would have completely missed the word "Family", then, both algorithms would have got the same scores, although the first one detected only 87% of the area of the "Family" GT text box.

DetEval *evaluation protocol*. DetEval [59] is a tool, based on the method of Wolf *et al.* [4], that is the core evaluation protocol used during the ICDAR 2011 and 2013 Robust reading competitions.

The system's object matching criteria can be configurable through eight parameters: six of them representing the minimum recall and precision overlap area between detections and GT objects for one-to-one, one-to-many and many-to-one cases, one parameter to add a border verification or not, and a threshold on the center distance between two matched boxes. We first evaluate the text detection results using a *"relaxed"* version of DetEval by



(a) (b) (c) (d) (e) (f)

Figure 13: Examples of detections; GT (red rectangles) and the detections (solid green rectangles); (a)-(c) one-to-one partial detections; (d)-(e) many-to-one detections. Corresponding scores are given in Table 4.

disabling the minimum area coverage constraints:

- the recall and precision area thresholds are set to 0;
- the center difference threshold is set to 1.

We have chosen this parametrization because it is in spirit the closer to our evaluation protocol, which is more permissive.

Next, we will describe the behavior of the *"relaxed"* DetEval protocol when dealing with one-to-one and many-to-one cases. Figure 13 shows some examples of partial one-to-one detections which got maximum recall scores, as seen in Table 4. However, the many-to-one detections from Figure 13, although they match entirely all GT text objects, are penalized, as seen in Table 4. Moreover, both recall and precision values are penalized and set to value 0.8 independently of the number of matched GT text boxes. Our method correctly penalizes the recall of one-to-one detections, and does not penalize the many-to-one cases. Hence, even when using the most permissive configuration of DetEval, our method is able to give better results in the evaluation of detections.

DetEval also integrates a set of new metrics to characterize both the quality and the quantity natures of a detector's output. Recall and precision are computed over a range of 20 different area threshold values to produce two curves. Then, two overall metrics are derived by computing their area under the curve (AUC). While these metrics solve the problem of partial matchings of the *"relaxed"* and default DetEval, the precision tends to even out the recall values when dealing with one-to-one cases (see Table 4), failing to differentiate the two characteristics of a detection. We have a similar problem

Table 4: Detection scores obtained with DetEval, ICDAR2003 and EvaLTex metrics for cases of Figure 13.

Image	Method	Recall	Precision	F-Score
Fig. 13a	DetEval (Relaxed)	1	1	1
	DetEval (AUC)	0.27	0.27	0.27
	ICDAR2003	0.74	0.74	0.74
	EvaLTex	0.61	1	0.75
Fig. 13b	DetEval (Relaxed)	1	1	1
	DetEval (AUC)	0.51	0.51	0.51
	ICDAR2003	0.64	0.64	0.64
	EvaLTex	0.63	1	0.77
Fig. 13c	DetEval (Relaxed)	1	1	1
	DetEval (AUC)	0.25	0.25	0.25
	ICDAR2003	0.7	0.7	0.7
	EvaLTex	0.55	1	0.7
Fig. 13d	DetEval (Relaxed)	0.8	0.8	0.8
	DetEval (AUC)	0.65	0.66	0.65
	ICDAR2003	0.35	0.39	0.36
	EvaLTex	1	1	1
Fig. 13e	DetEval (Relaxed)	0.8	0.8	0.8
	DetEval (AUC)	0.73	0.74	0.73
	ICDAR2003	0.54	0.77	0.63
	EvaLTex	1	1	1
Fig. 13f	DetEval (Relaxed)	0.8	0.8	0.8
	DetEval (AUC)	0.72	0.72	0.72
	ICDAR2003	0.61	0.65	0.62
	EvaLTex	1	1	1

for the many-to-one matches (see Table 4): the small difference between the recall and precision values does not give a clear distinction between the number of GT boxes that were detected and how well these detections were covering them.

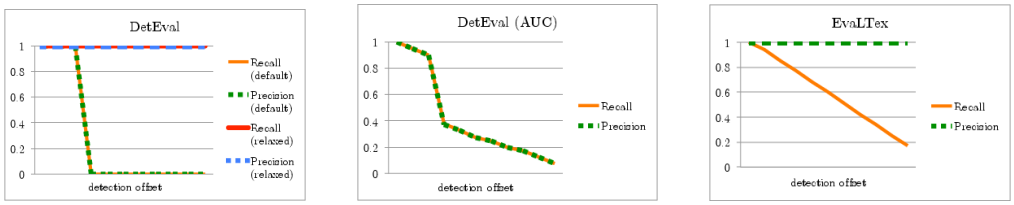
ICDAR2003 ***evaluation protocol***. The ICDAR2003 protocol, also used during the ICDAR 2005 competition, is still widely used for evaluating text localization methods [60, 61, 62, 63, 64, 65]. The advantage of this method is that the recall for partial one-to-one matches is scored accordingly to the true ratio between the intersection and the GT surface. On the other hand, the

precision is not computed with respect to the detection surface and, similarly to DetEval protocol, seems to always be equal to the recall rate, as it can be seen in Table 4.

Another main drawback is due to the choice of the best match approach used to solve the many-to-one cases, illustrated in Figure 13, whose corresponding scores are given in Table 4.

Evaluating the one-to-one detection. As previously said, one-to-one detections are treated differently from one protocol to another one. We propose a simple experiment which consists in gradually decreasing the quality (coverage area) of a one-to-one detection (see Figure 14a) and analyze the recall and precision evolution depending on this.

(a) A sequence of one-to-one detections.



(b) (c) (d)

Figure 14: Recall and precision plots for one-to-one detection cases using different evaluation protocols; (a) the covering area between the detection (in green) and the GT object (word "ROUTE") is reduced; (b) DetEval (default and relaxed); (c) DetEval (AUC); (d) EvaLTex.

Figure 14b shows the evolution of recall and precision scores given by the default and *"relaxed"* configuration of DetEval when dealing with par-

tial one-to-one matchings. For the *"relaxed"* DetEval recall and precision values are constant and equal 1, because of area thresholds set to 0. On the other hand, for the default DetEval, there is an irregular decreasing of recall and precision values. This reflects the binary evaluation of one-to-one matchings, which depends on the recall and precision area parameters and which can not correctly differentiate total and partial detections.

Figure 14c illustrates the behavior of DetEval AUC metrics. One can observe that we have similar plots for recall and precision. However, since the intersection area always remains within the boundaries of the GT, the metrics should have a different behavior. Our method (Figure 14d), evaluates the precision to a fix value of 1, which is a logical because the detection correctly fits the GT object. On the contrary, the recall score decreases linearly with the progressive shrinkage of the detection box.

ICDAR *2013 Robust Reading competition results*. We now consider the detection results of ten participants at the ICDAR 2013 Robust Reading competition (*Challenge 2*) [44]. Our goal is to compare scores given by different evaluation protocols on the same dataset as in this competition. Table 5 gives the scores obtained for the ten participants with the ICDAR 2003 protocol [31], DetEval [44], DetEval AUC and EvaLTex. The results of these methods are publicly available on the competition website page [44].

Among the four protocols, the DetEval AUC metrics seem to be the strictest. But, even if the ICDAR evaluation protocol is less penalizing than the AUC metrics, in the case of the *Inkam* participant, it gives lower scores. This is because a high number of partial one-to-one detections are rejected

Table 5: Detection scores and rankings of all participants during the ICDAR 2013 Robust Reading Competition (Challenge 2) using the ICDAR2003 [31], DetEval [44] and EvaLTex evaluation protocols. ↑, ↓ and ∼ symbols are used to depict the tendency of scores produced with the four methods (DetEval scores with respect to ICDAR2003, DetEval AUC scores with respect to DetEval, and EvaLTex scores with respect to DetEval AUC).

	Recall				Precision				FScore *(Ranking)*			
Method	ICDAR 2003	DetEval	DetEval (AUC)	EvaLTex	ICDAR 2003	DetEval	DetEval (AUC)	EvaLTex	ICDAR 2003	DetEval	DetEval (AUC)	EvaLTex
USTB_TexStar	0.58	0.66 ↑	0.69 ↑	0.72 ↑	0.80	0.88 ↑	0.89 ↑	0.93 ↑	0.67 (1)	0.76 (1)	0.78 (1)	0.82 (1)
TextSpotter	0.49	0.65 ↑	0.65 ∼	0.66 ↑	0.69	0.88 ↑	0.87 ↓	0.77 ↓	0.57 (6)	0.74 (2)	0.75 (2)	0.71 (8)
CASIA_NLPR	0.55	0.68 ↑	0.69 ↑	0.73 ↑	0.67	0.79 ↑	0.79 ∼	0.87 ↑	0.61 (5)	0.73 (3)	0.74 (4)	0.80 (4)
Text_detector_CASIA	0.54	0.63 ↑	0.67 ↑	0.72 ↑	0.75	0.85 ↑	0.85 ∼	0.91 ↑	0.63 (4)	0.72 (4)	0.75 (2)	0.80 (4)
I2R_NUS_FAR	0.62	0.69 ↑	0.71 ↑	0.76 ↑	0.71	0.75 ↑	0.76 ↑	0.88 ↑	0.66 (2)	0.72 (4)	0.73 (6)	0.82 (1)
I2R_NUS	0.61	0.66 ↑	0.70 ↑	0.75 ↑	0.69	0.73 ↑	0.73 ∼	0.86 ↑	0.65 (3)	0.69 (6)	0.72 (6)	0.81 (3)
TH-TextLoc	0.53	0.65 ↑	0.70 ↑	0.74 ↑	0.58	0.70 ↑	0.70 ∼	0.74 ↑	0.55 (7)	0.67 (7)	0.70 (7)	0.74 (7)
Text Detection	0.49	0.53 ↑	0.66 ↑	0.74 ↑	0.62	0.74 ↑	0.74 ∼	0.87 ↑	0.55 (7)	0.62 (8)	0.70 (7)	0.80 (4)
Baseline	0.30	0.35 ↑	0.35 ∼	0.36 ↑	0.56	0.61 ↑	0.61 ∼	0.63 ↑	0.39 (9)	0.44 (9)	0.45 (9)	0.46 (10)
Inkam	0.37	0.35 ↓	0.43 ↑	0.55 ↑	0.32	0.31 ↓	0.32 ↑	0.57 ↑	0.34 (10)	0.33 (10)	0.36 (10)	0.56 (9)

by ICDAR, due to its covering area constraint. On the contrary, AUC metrics, by varying this constraint, can better handle partial matchings. An interesting point is the similarity of the ranking produced by AUC metrics and EvaLTex, despite of their high score variations.

While DetEval recall scores are 13% higher compared to ICDAR scores (see the *Text Detection* participant), their precision score is similar. This high recall difference can be explained by the large number of GT objects involved in many-to-one matchings, that are rejected by the ICDAR protocol but not by DetEval. Furthermore, EvaLTex recall scores tend to be higher than those of ICDAR and DetEval protocols, because the former fairly validates more partial one-to-one matchings. Moreover, EvaLTex relaxes the unfair precision penalties applied by the other two methods. On the contrary, it penalizes algorithms that produce detection areas significantly larger than the GT objects (Figure 15), as in the case of *TextSpotter* participant, which gets a precision score 10% lower compared to DetEval score. On the other

Figure 15: *TextSpotter* detection examples.

hand, it gives higher precision scores to algorithms *I2R_NUS_FAR*, *I2R_NUS* and *Inkam* for which a high number of partial one-to-one detections were

mismatched by DetEval and ICDAR protocols. To finish, we note that ICDAR and DetEval rankings are relatively similar, while EvaLTex proposes substantial changes in it.

4.1.2. Region annotation: impact on global scores

Figure 16 shows the impact of the GT annotation on the precision and recall scores (computed in Section 3.3). One can observe that the precision value increases proportionally to the surface of the text region. This is logical because the more GT objects a region contains, the smaller the non textual area becomes and therefore the less the precision is penalized.

(a) (b) $R = 1, P = 0.60$ (c) $R = 1, P = 0.62$ (d) $R = 1, P = 0.65$

Figure 16: The impact of the region GT (yellow rectangles) annotation on precision and recall scores; (a) 3 GT objects (red rectangles), 1 detection (green filled rectangle); (b) 3 GT objects grouped into 3 text regions; (c) 3 GT objects grouped into 2 text regions; (d) 3 GT objects grouped in one text region.

Table 6 shows the interest of using the two-level annotation on some key examples in Figure 17. Most of the detections correspond to many-to-one matchings. Here, the region labeling is done at line level. As it can be seen, by enabling this region annotation (and then having a two-level annotation), we get higher precision scores. On the contrary, recall scores are not changed.

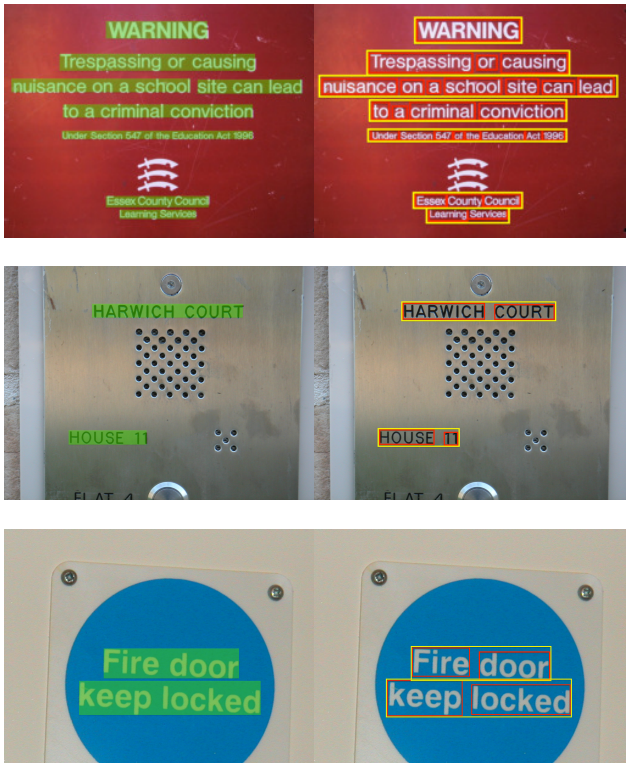


Figure 17: Left: detections (green filled rectangles); right: object GT annotation (red rectangles) and region annotation (yellow rectangles).

4.2. Experimental results using the mask annotation

In this subsection we compare the evaluation given by EvaLTex to detection cases depending on the type of annotation (rectangles or masks). Figure 18 shows most of the problems that may be encountered when using a rectangular annotation: GT rectangles that contain more non-textual areas (Figure 18b, 18c, 18d, 18e, 18g), intersection of boxes in the GT (Figure 18d, 18f, 18g), inclusions of GT boxes (Figure 18f).

The corresponding evaluation scores are presented in Table 7. One can observe that, when using a rectangular representation, the matching is disturbed by the text objects that intersect in the GT. Namely, text objects such as "ALBACORE" in Figure 18d, are matched two times: with their corresponding detection and with detections targeting objects that intersect them in the GT. Hence, the coverage scores of such GT objects are penalized by the fragmentation parameter invoked during the one-to-many matching, which can furthermore impact the global recall score. This can however be avoided for similar cases, such as "GRAS" and "ANISETTE" (Figure 18f), by using the filtering procedure described in Section 3.2.1.

Table 6: Global scores, recall and precision, when enabling and disabling the region GT annotation.

	No GT region		GT region	
Fig. 17	**Recall**	**Precision**	**Recall**	**Precision**
Top	1	0.96	1	1
Middle	1	0.89	1	1
Bottom	1	0.92	1	1

Recall values of Figure 18e show another example of difference when using these two representations in the case of a tilted and perspective deformed

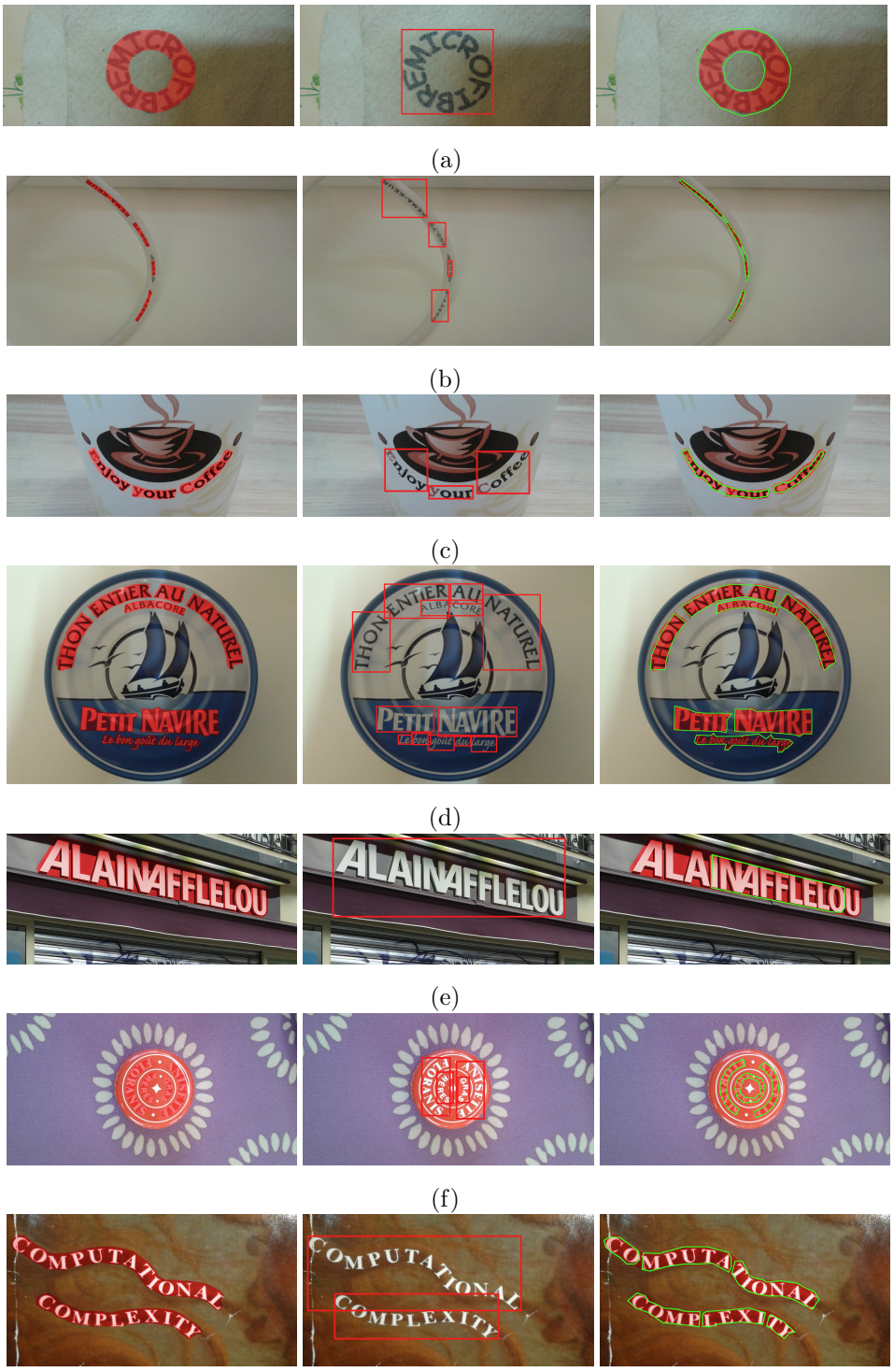


Figure 18: Examples of text areas (inclined, curved, in perspective, circular); left: mask GT annotation (red); center: rectangular GT annotation (red); right: GT masks (red) overlapped by detection masks (green). The rectangular detections are not represented but correspond to boxes surrounding the detection masks in the last column which are matched with the GT rectangles shown in the second column.

text ("ALAINAFFLELOU"), only partially matched. The coverage ratio computed with rectangles is smaller than the one computed with masks and consequently leads to a significant recall difference.

Another difference between these two annotation types comes from the precision variations that are higher when dealing with many-to-one detections. Such situations can be seen in Figures 18b and 18c that illustrate many-to-one detections covering curved text strings ("KEMA-KEUR", "3G0.75" and "VDE" in Figure 18b, respectively GT objects "Enjoy" and "yours" in Figure 18c). For Figure 18b, the precision values vary from 0.48, when using the rectangular representation to 0.81, in the case of mask annotation. Similarly, the precision scores for the two text representations in Figure 18c range from 0.73 to 0.98. Once again, the rectangle representation shows its limitation and that it can significantly penalize the evaluation of a detector.

5. Conclusions

Today, no accurate protocol allows a reliable evaluation and comparison of text detection algorithm outputs. The few existing protocols used in several challenges have many problems. We claim that it is of crucial importance for the researchers to evaluate and compare their detection algorithms using a strong and reliable protocol so that they can better evaluate the pros and cons of their algorithms and improve them.

That is why, in this paper, we have introduced a novel approach to evaluate and compare text localization algorithms, that overcomes some of the existing drawbacks of current evaluation systems. We can cite several systems: the best match approach, which assigns a many-to-one detection to

Table 7: Object (*Cov* and *Acc*) and global (*Recall*, *Precision* and *F*-Score) performance scores corresponding to detections in Figure 18 computed using the rectangular and the mask representations.

		Rectangular representation					Mask representation				
Figure	**GT text object**	**Cov**	**Acc**	**Recall**	**Precision**	***F*-Score**	**Cov**	**Acc**	**Recall**	**Precision**	***F*-Score**
18a	*MICROFIBRE*	1	1	1	1	1	1	1	1	1	1
18b	*KEMA-KEUR*	1	0.32	1	0.48	0.65	0.97	0.76	0.96	0.81	0.88
	3G0.75	1	0.3				0.97	0.76			
	VDE	1	0.32				1	0.75			
	H03VV-F	1	1				0.9	0.98			
18c	*Enjoy*	1	0.6	1	0.73	0.85	0.98	0.96	0.94	0.98	0.96
	your	1	0.6				0.9	0.96			
	Coffee	1	1				0.93	1			
18d	*THON*	1	1	0.97	0.88	0.92	1	1	1	0.9	0.94
	ENTIER	1	0.98				1	0.95			
	AU	1	0.98				0.99	0.95			
	NATUREL	1	1				1	1			
	ALBACORE	0.59	1				1	0.92			
	PETIT	1	1				1	1			
	NAVIRE	1	1				1	1			
	Le	1	0.72				1	0.78			
	bon	1	0.73				1	0.78			
	gout	1	0.72				1	0.78			
	du	1	0.71				1	0.78			
	large	1	0.73				1	0.8			
18e	*ALAINAFFLELOU*	0.58	1	0.58	1	0.73	0.63	1	0.63	1	0.77
18f	*FLORANIS*	1	1	1	0.93	0.96	0.96	1	0.97	0.91	0.94
	ANISETTE	1	1				0.97	1			
	GRAS	1	0.85				0.97	0.83			
	FRERES	1	0.86				0.96	0.83			
18g	*COMPUTATIONAL*	0.34	1	0.28	1	0.44	0.46	0.99	0.45	0.99	0.63
	COMPLEXITY	0.31	1				0.45	0.99			

only one GT object and rejects other valid matched GT text boxes ; the minimum overlap area constraint which assigns to a detection box a GT object if and only if their intersection area is large enough ; the use of inappropriate GT annotations, or also the lack of distinction between recall and precision scores when dealing with partial detections. Even if it would seem reasonable for many object detection purposes, for text detection these constraints are rather restrictive and usually lead to severe penalties. The purpose of our evaluation is not to provide detections with higher scores, but more precise ones that reflect more accurately the reality.

The novelty consists in the definition of a set of new rules and the reinterpretation of standard metrics at object level, coverage and accuracy, to improve the evaluation quality. When using a rectangular text representation we introduce a new GT granularity level, the region tag, to relax the precision penalizations and to allow an evaluation of word and line-level outputs. Moreover, our evaluation protocol identifies and treats independently the one-to-one, one-to-many, many-to-one and many-to-many matches. The protocol penalizes the recall in cases of fragmented detections, and penalizes the precision if the detections are not accurate enough. Furthermore, we proposed quality and quantity performance measures that can capture the whole complexity of a detection. Global recall and precision metrics are then obtained by combining the quality and quantity values. Finally, we proved that our evaluation framework can handle different GT annotation and detection representations, such as polygonal, elliptical or free-form shapes. Consequently, our procedure provides a more realistic and representative evaluation comparison between different text detection algorithms.

Notice that our evaluation protocol can be seen as a first step of an truthful end-to-end evaluation protocol because, in order to compare the text transcriptions with the GT, a reliable matching strategy is required. Without a matching procedure, robust to granularity differences, systems would be under evaluated, and hence many transcriptions would not be compared to the correct GT objects.

Further work will consist in adapting the two-level option used for rectangle GT annotation to mask representation. This task requires a precise annotation algorithm and the definition of a procedure that permits to link and group GT objects into mask regions. An additional work will focus on evaluating the results of text detection algorithms on more challenging datasets, such as the Street View Text, iTowns and MSRA-TD500.

Acknowledgment

This work was partially supported by FUI 14 (LINX project).

References

[1] Y. Qixiang, D. Doermann, Text detection and recognition in imagery: A survey, Pattern Analysis and Machine Intelligence 37 (7) (2015) 1480–1500.

[2] J. Fabrizio, B. Marcotegui, M. Cord, Text detection in street level image, Pattern Analysis and Applications 16 (4) (2013) 519–533.

[3] L. Neumann, J. Matas, A method for text localization and recognition in real-world images, in: Asian Conference on Computer Vision, 2010, pp. 770–783.

[4] C. Wolf, J.-M. Jolion, Object count/area graphs for the evaluation of object detection and segmentation algorithms, International Journal on Document Analysis and Recognition 8 (4) (2006) 280–296.

[5] L. Sun, Q. Huo, W. Jia, K. Chen, Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks, in: International Conference on Pattern Recognition, 2014, pp. 2715–2720.

[6] C. Merino-Gracia, K. Lenc, M. Mirmehdi, A head-mounted device for recognizing text in natural scenes., in: International Workshop on Camera-Based Document Analysis and Recognition, 2011, pp. 29–41.

[7] Y.-F. Pan, X. Hou, C.-L. Liu, Text localization in natural scene images based on conditional random field, in: International Conference on Document Analysis and Recognition, 2009, pp. 6–10.

[8] L. Sun, Q. Huo, W. Jia, K. Chen, A robust approach for text detection from natural scene images, Pattern Recognition 48 (9) (2015) 2906 – 2920.

[9] Y. Du, G. Duan, H. Ai, Context-based text detection in natural scenes, in: International Conference on Image Processing, 2012, pp. 1857–1860.

[10] L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Conference on Computer Vision and Pattern Recognition, 2012, pp. 3538–3545.

[11] W. Huang, Y. Qiao, X. Tang, Robust scene text detection with convolution neural network induced mser trees, in: European Conference on Computer Vision, 2014, pp. 497–511.

[12] L. Kang, Y. Li, D. Doermann, Orientation robust text line detection in natural images, in: Conference on Computer Vision and Pattern Recognition, 2014, pp. 4034–4041.

[13] C. Yao, X. Bai, W. Liu, A unified framework for multioriented text detection and recognition, Transactions on Image Processing 23 (11) (2014) 4737–4749.

[14] X. Huang, H. Ma, Automatic detection and localization of natural scene text in video, in: International Conference on Pattern Recognition, 2010, pp. 3216–3219.

[15] J. Zhang, R. Kasturi, Sign detection based text localization in mobile device captured scene images, in: International Workshop on Camera-Based Document Analysis and Recognition, 2014, pp. 71–82.

[16] P. Tomer, A. Goyal, Ant clustering based text detection in natural scene images, in: International Conference on Computing, Communications and Networking Technologies, 2013, pp. 1–7.

[17] X. Wang, Y. Song, Y. Zhang, Natural scene text detection with multi-channel connected component segmentation, in: International Conference on Document Analysis and Recognition, 2013, pp. 1375–1379.

[18] S. Liu, Y. Zhou, Y. Zhang, Y. Wang, W. Lin, Text detection in natural

scene images with stroke width clustering and superpixel, in: Advances in Multimedia Information Processing, 2014, pp. 123–132.

[19] Y. Li, C. Shen, W. Jia, A. van den Hengel, Leveraging surrounding context for scene text detection, in: International Conference on Image Processing, 2013, pp. 2264–2268.

[20] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, C. Koch, Adaboost for text detection in natural scene, in: International Conference on Document Analysis and Recognition, 2011, pp. 429–434.

[21] C. Yi, Y. Tian, Text detection in natural scene images by stroke gabor words, in: International Conference on Document Analysis and Recognition, 2011, pp. 177–181.

[22] C. Yi, Y. Tian, Assistive text reading from complex background for blind persons, in: International Workshop on Camera-Based Document Analysis and Recognition, 2012, pp. 15–28.

[23] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, Scene text detection using graph model built upon maximally stable extremal regions, Pattern Recognition Letters 34 (2) (2013) 107 – 116.

[24] D. Kumar, M. N. A. Prasad, A. G. Ramakrishnan, Multi-script robust reading competition in icdar 2013, in: International Workshop on Multilingual OCR, 2013, pp. 14:1–14:5.

[25] S. Lu, T. Chen, S. Tian, J.-H. Lim, C.-L. Tan, Scene text extraction based on edges and support vector regression, International Journal on Document Analysis and Recognition 18 (2) (2015) 125–135.

[26] C. Shi, C. Wang, B. Xiao, S. Gao, J. Hu, End-to-end scene text recognition using tree-structured models, Pattern Recognition 47 (9) (2014) 2853 – 2866.

[27] X.-C. Yin, X. Yin, K. Huang, H.-W. Hao, Robust text detection in natural scene images, Pattern Analysis and Machine Intelligence 36 (5) (2014) 970–983.

[28] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Reading text in the wild with convolutional neural networks, International Journal on Computer Vision (2015) 1–20.

[29] K. Zagoris, I. Pratikakis, Text detection in natural images using bio-inspired models, in: International Conference on Document Analysis and Recognition, 2013, pp. 1370–1374.

[30] Q. Ye, D. Doermann, Scene text detection via integrated discrimination of component appearance and consensus, in: International Workshop on Camera-Based Document Analysis and Recognition, 2014, pp. 47–59.

[31] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, ICDAR 2003 robust reading competitions, in: International Conference on Document Analysis and Recognition, 2003, pp. 682–687.

[32] S. M. Lucas, ICDAR 2005 text locating competition results, in: International Conference on Document Analysis and Recognition, 2005, pp. 80–84.

[33] X.-S. Hua, L. Wenyin, H.-J. Zhang, Automatic performance evaluation for video text detection, in: International Conference on Document Analysis and Recognition, 2001, pp. 545–550.

[34] J. Weinman, E. Learned-Miller, A. Hanson, Scene text recognition using similarity and a lexicon with sparse belief propagation, Pattern Analysis and Machine Intelligence 31 (10) (2009) 1733–1746.

[35] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Conference on Computer Vision and Pattern Recognition, 2010, pp. 2963–2970.

[36] I. Posner, P. Corke, P. Newman, Using text-spotting to query the world, in: Intelligent Robots and Systems, 2010, pp. 3181–3186.

[37] S. Lee, M. S. Cho, K. Jung, J. H. Kim, Scene text extraction with edge constraint and text collinearity, in: International Conference on Pattern Recognition, 2010, pp. 3983–3986.

[38] K. Wang, S. Belongie, Word spotting in the wild, in: European Conference on Computer Vision, 2010, pp. 591–604.

[39] T. Q. Phan, P. Shivakumara, S. Tian, C. L. Tan, Recognizing text with perspective distortion in natural scenes, in: International Conference on Computer Vision, 2013, pp. 569–576.

[40] R. Nagy, A. Dicker, K. Meyer-Wegener, Neocr: A configurable dataset for natural image text recognition, in: International Workshop on Camera-Based Document Analysis and Recognition, 2012, pp. 150–163.

[41] C. Yi, Y. Tian, Text string detection from natural scenes by structure-based partition and grouping, Transactions on Image Processing 20 (9) (2011) 2594–2605.

[42] A. Shahab, F. Shafait, A. Dengel, Icdar 2011 robust reading competition challenge 2: Reading text in scene images, in: International Conference on Document Analysis and Recognition, 2011, pp. 1491–1496.

[43] C. Yao, Detecting texts of arbitrary orientations in natural images, in: Conference on Computer Vision and Pattern Recognition, 2012, pp. 1083–1090.

[44] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. de las Heras, ICDAR 2013 robust reading competition, in: International Conference on Document Analysis and Recognition, 2013, pp. 1484–1493, http://dag.cvc.uab.es/icdar2013competition/.

[45] A. Risnumawan, P. Shivakumara, C. S. Chan, C. L. Tan, A robust arbitrary text detection system for natural scene images, Expert Systems with Applications 41 (18) (2014) 8027–8048.

[46] A. Clavelli, D. Karatzas, J. Lladós, A framework for the assessment of text extraction algorithms on complex color images, in: International Workshop on Document Analysis Systems, 2010, pp. 19–26.

[47] V. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, H. Li, D. Doermann, T. Drayer, Performance evaluation of object detection algorithms, in: International Conference on Pattern Recognition, 2002, pp. 965–969.

[48] J. Nascimento, J. Marques, Performance evaluation of object detection algorithms for video surveillance, Transactions on Multimedia 8 (4) (2006) 761–774.

[49] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, X. Lin, Icdar 2003 robust reading competitions: entries, results, and future directions, International Journal on Document Analysis and Recognition 7 (2-3) (2005) 105–122.

[50] Y. Ma, C. Wang, B. Xiao, R. Dai, Usage-oriented performance evaluation for text localization algorithms, in: International Conference on Document Analysis and Recognition, 2007, pp. 1033–1037.

[51] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, J. Zhang, Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol, Pattern Analysis and Machine Intelligence 31 (2) (2009) 319–336.

[52] M. Anthimopoulos, B. Gatos, I. Pratikakis, A two-stage scheme for text detection in video images, Image and Vision Computing 28 (9) (2010) 1413 – 1426.

[53] P. Shivakumara, T. Q. Phan, C. Tan, A gradient difference based technique for video text detection, in: International Conference on Document Analysis and Recognition, 2009, pp. 156–160.

[54] P. Shivakumara, H. Basavaraju, D. Guru, C. Tan, Detection of curved text in video: Quad tree based method, in: International Conference on Document Analysis and Recognition, 2013, pp. 594–598.

[55] P. Shivakumara, T. Q. Phan, C. Tan, A robust wavelet transform based technique for video text detection, in: International Conference on Document Analysis and Recognition, 2009, pp. 1285–1289.

[56] P. Shivakumara, T. Q. Phan, C. Tan, A laplacian approach to multi-oriented text detection in video, Pattern Analysis and Machine Intelligence 33 (2) (2011) 412–419.

[57] Maurdor, Evaluation plan, `www.maurdor-campaign.org`.

[58] S. Mao, T. Kanungo, Software architecture of pset: a page segmentation evaluation toolkit., International Journal on Document Analysis and Recognition 4 (3) (2002) 205–217.

[59] Deteval - evaluation software for object detection algorithms, `http://liris.cnrs.fr/christian.wolf/software/deteval/`.

[60] B. Bai, F. Yin, C. L. Liu, Scene text localization using gradient local correlation, in: Document Analysis and Recognition, 2013, pp. 1380–1384.

[61] S. Karaoglu, B. Fernando, A. Tremeau, A novel algorithm for text detection and localization in natural scene images., in: International Conference on Digital Image Computing: Techniques and Applications, 2010, pp. 635–642.

[62] S. Karaoglu, J. Gemert, T. Gevers, Object reading: Text recognition for object recognition, in: European Conference on Computer Vision, 2012, pp. 456–465.

[63] H. I. Koo, D. H. Kim, Scene text detection via connected component clustering and nontext filtering, Transactions on Image Processing 22 (6) (2013) 2296–2305.

[64] J. Mao, H. Li, W. Zhou, S. Yan, Q. Tian, Scale based region growing for scene text detection, in: International Conference on Multimedia, 2013, pp. 1007–1016.

[65] T. Q. Phan, P. Shivakumara, C. L. Tan, Detecting text in the real world., in: ACM Conference on Multimedia, 2012, pp. 765–768.