Multi-script robust reading competition in ICDAR 2013

Deepak Kumar
MILE Laboratory
Dept. of EE
IISc, Bengaluru, INDIA
deepak@ee.iisc.ernet.in

M N Anil Prasad
MILE Laboratory
Dept. of EE
IISc, Bengaluru, INDIA
anilprasadmn@ee.iisc.ernet.in

A G Ramakrishnan
MILE Laboratory
Dept. of EE
IISc, Bengaluru, INDIA
ramkiag@ee.iisc.ernet.in

ABSTRACT

A competition was organized by the authors to detect text from scene images. The motivation was to look for script-independent algorithms that detect the text and extract it from the scene images, which may be applied directly to an unknown script. The competition had four distinct tasks: (i) text localization and (ii) segmentation from scene images containing one or more of Kannada, Tamil, Hindi, Chinese and English words. (iii) English and (iv) Kannada word recognition task from scene word images. There were totally four submissions for the text localization and segmentation tasks. For the other two tasks, we have evaluated two algorithms, namely nonlinear enhancement and selection of plane and midline analysis and propagation of segmentation, already published by us. A complete picture on the position of an algorithm is discussed and suggestions are provided to improve the quality of the algorithms. Graphical depiction of f-score of individual images in the form of benchmark values is proposed to show the strength of an algorithm.

Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture

Keywords

Multi-script, Robust reading competition, Text localization, Text segmentation, Scene images, English word recognition, Kannada word recognition, PLT, MAPS, NESP, Benchmark

1. INTRODUCTION

A decade has passed since the first robust reading competition (RRC) on camera-captured scene images was organized by Lucas et. al in ICDAR 2003 [12]. Subsequent competitions were held in ICDAR 2005 and 2011 by Lucas et. al [13] and Shahab et. al [16], respectively. Born-digital images were introduced in RRC by Karatzas et. al [3] in ICDAR 2011. However, no competition has been held on multi-script camera-captured scene images; there are a few research contributions though [5, 11]. Applications which transcribe or translate words of unknown scripts in a scene are of great value to a foreigner visitor.

MOCR '13, August 24 2013, Washington, DC, USA

http://dx.doi.org/10.1145/2505377.2505390

Figure 1: Sample multi-script images provided for training in the text localization and segmentation tasks of the competition.

In the Indian scenario, we find hoardings and street name boards in multiple languages (scripts). A multi-script robust reading competition (MRRC) is organized, as part of ICDAR 2013 [2], to motivate the development of novel applications for identification and recognition of Indic scripts in camera-captured scene images. The images contain text in one of Roman, Kannada, Tamil, Devanagari and Chinese scripts. Figure 1 shows some sample images from the training set of text localization and segmentation tasks. This MRRC gave a platform for researchers around the globe to address this issue, hitherto very less explored. The competition ran in open mode, where each participant downloaded the test set and uploaded the results of their algorithms.[1]. Thirty people registered to participate in MRRC and three of them submitted their results.

[1] http://mile.ee.iisc.ernet.in/mrrc

(a) English word images

(a) Kannada word images

Figure 2: Sample word images used for English and Kannada word recognition tasks in MRRC.

2. DATASETS COLLECTED FOR MRRC

We collected nearly 4000 camera-captured images mainly from Bengaluru city roads, Karnataka, India. Four different tasks were organized in this competition, namely

1. Text localization: Obtain a bounding box around the text, irrespective of the script.
2. Text segmentation: Identify the text pixels, irrespective of the script.
3. Word recognition: Recognize the words from the given set of manually segmented word images, containing:
   (a) English words + Indo-Arabic numerals
   (b) Kannada words

In this competition, 167 camera-captured scene images each were provided for training and testing (1:1) for text localization and segmentation tasks. 495 and 645 word images were provided as the training and test set, respectively, for English word recognition task. Kannada word recognition task had 300 training and 243 test images. Figure 2 shows some samples from the English and Kannada training set. All the images had a background of two pixels all around the located boundary to provide proper background. Ground-truth for the data set was created using our multi-script annotation toolkit (MAST) [4, 14], a user interface (available for free download) to annotate scene images at the pixel level.

We name our data set as 'Multi-script and scene text reading' (MASTER) data set. An exhaustive variety of degradations and challenges are covered in this dataset, namely artistic, curve, depth, emboss, engrave, glossy, handwritten, illumination, motion blur, multi-colored, multi-font, multi-script, night-vision, normal, occlusion, resolution, shading, shear and slant. An image captured during night with normal mode has been included in the dataset as a night-vision sample, for the first time.

3. EVALUATION PROCEDURE

Lucas et. al proposed a procedure for algorithm evaluation in ICDAR 2003 competition [12]. The same was used in ICDAR 2005 competition [13]. Wolf and Jolion [18] proposed another method to evaluate the algorithms on text localization task, since [12] heavily penalizes algorithms for detecting text lines. We use Wolf and Jolion method to evalaute the algorithms on text localization task. The area recall and precision thresholds are $t_r = 0.8$ and $t_p = 0.4$, respectively. For one-to-many matches, $f_{cs}(k) = 0.8$ is used for $Match_G$ calculation and for many-to-one matches, $f_{cs}(k) = 1$ is used for $Match_D$ calculation [18]. The $f_{cs}(k)$ values indicate punishing over-segmented words and no punishment for under-segmented words, which in turn reduces penalization on the text line locating algorithms.

Text segmentation performance is usually evaluated [1] using connected components (CCs). Ground-truth CCs of a test image are matched against the output of the algorithm to determine whether the components are well-segmented, merged, broken or lost. This evaluation does not account for non-text components output by an algorithm. Hence, we employ pixel-level information to evaluate the algorithms. Precision, recall and f-score are calculated for the participating algorithms on the text segmentation task.

Word recognition results are evaluated based on the number of correctly recognized words and Levenshtein distance computed from Unicode strings.

Precision, recall and f-score values of an algorithm on an entire dataset do not reveal any information about the algorithm performance on individual images. Hence, a novel method is proposed to evaluate text localization and segmentation algorithms graphically.

Completion calls for presenting results for all the tasks of a competition. Hence, we have also obtained results on the test images using our own algorithms to be used as the baseline for the other tasks conducted in the competition.

4. METHODS SUBMITTED

We had participation from three different countries: Spain, China and India. A brief description of all the methods submitted is provided in the following sub-sections.

4.1 Text localization, segmentation by Yin et. al[2]

The character candidates are extracted by exploring the hierarchical structure of maximally stable extremal regions (MSERs) and adopting simple features. They are clustered into text candidates (TC) by a single-link algorithm, where distance weights and threshold for clustering are learned automatically by a novel, self-training, distance metric learning method. The posterior probabilities of TC corresponding to non-text are estimated with a character classifier using Bayes' rule; TC with high non-text probabilities are eliminated and finally texts are identified using a text classifier.

[2]Xuwang Yin and Xu-Cheng Yin, Dept. of Comp Science and Tech, School of Comp and Commun Engg, Univ of Science and Tech, Beijing and Hong-Wei Hao, Inst. of Automation, Chinese Academy of Sciences.

Text candidates are in fact MSERs, which are sets of pixels, text is segmented by setting pixels presented in text as white (text pixels) and others as black.

4.2 Text segmentation by Gómez et. al[3]

In the preprocessing stage, MSER algorithm is used to obtain a region decomposition of the input image. Then, two different clustering techniques are combined in a single parameter-free procedure to detect groups of regions organized as text. The maximally meaningful groups are first detected in several feature spaces, where each feature space is a combination of proximity information (x,y coordinates) and a similarity measure (intensity, color, size, gradient magnitude, etc.), thus providing a set of hypotheses of text groups. Evidence accumulation framework is used to combine all these hypotheses to get the final estimate. The resulting method is independent of the script, can deal with any kind of font types and sizes, and is not constrained to horizontally aligned text.

4.3 Text segmentation by Sethi et. al[4]

In this method, natural scene images are segmented for text based on edge analysis and morphological operators. The images are converted to gray scale and Canny edges are detected. The edge image is morphologically dilated and analyzed to remove edges corresponding to non-text regions. Then, the image is binarized using the mean and standard deviation values of edge pixels. The resulting image is post-processed to fill the gaps and smoothen the text strokes.

4.4 Baseline methods

The following algorithms, already published by the authors, have been applied on the data to obtain reference level (baseline) performance for comparison:

Text segmentation algorithm (OTCYMIST): Scene images are scaled to a standard height of 320 pixels preserving the aspect ratio. R, G and B color channels of the image are binarized separately using Otsu's threshold. CCs of binarized color plane and its complement are labeled. The CCs are filtered based on area, aspect ratio and Euler number. The filtered CCs from each binarized plane and its complement are morphologically thinned and combined to form a single thinned plane [9]. Thinned images are rescaled to the original size for subsequent processing.

Each CC from a thinned plane forms a node of a graph and a minimum spanning tree is constructed. Edges longer than 2.5 times mean edge length are removed. Isolated nodes are removed and the remaining nodes are compared for height consistency. The pruning process is repeated once. The CCs preserved by all three thinned planes are merged. The filtered CCs in the single merged image plane are grouped horizontally to obtain a bounding box (BB). Overlapping BBs are filtered out. Non-overlapping and partially overlapping BBs with the CCs form the segmentation result [9].

Word recognition algorithms (PLT/NESP/MAPS/ Benchmark): Cropped scene word images with height less than 60 pixels are scaled up by three times; those with height exceeding 180 pixels are scaled down to a height of 180 pixels and remaining images are unaltered. This preprocessing is common to PLT, NESP and MAPS algorithms.

PLT: The gray scale image obtained from preprocessing is enhanced by applying power-law transform and then segmented using Otsu's method [10].

NESP: Fischer discrimination factor is calculated for red, green, blue, intensity and lightness planes after enhancement by different power-law values [8]. The plane with maximum discrimination value is selected for segmentation.

MAPS: The middle row pixels of the gray scale image are segmented into two classes using the minimum and maximum values in a window [7]. Mean and variance of the resulting classes are used to segment the non-middle row pixels with neighborhood criteria.

Benchmark: The pixel-level segmented test word images are extracted using the BB information from the annotated (using MAST) text localization results. These clean images are used to benchmark [6] the word recognition rate on the MASTER data set.

The word images segmented by each of the above algorithms [6, 7, 8, 10] are padded with zeros around the boundary, with the minimum of the image height or width value. They are then fed to Omnipage or Tesseract OCR [15, 17], for English or Kannada word recognition, respectively.

5. RESULTS AND DISCUSSION

The results of the participants' algorithms, as well as the reference algorithms, are discussed below, for each task:

5.1 Text localization task

A novel method is introduced to analyze text locating and segmenting algorithms. Let GT_i be the fraction of ground-truth text pixels in the i^{th} image. All the images in the dataset are sorted based on their GT_i values. Two metrics, called benchmark (B_i) and algorithm result (AR_i) are computed for the individual images using inverse grading:

$$B_i = 1/GT_i \tag{1}$$

$$AR_{ji} = F_{ji}/GT_i \tag{2}$$

where, F_{ji} and AR_{ji} are the f-score and the result of the j^{th} algorithm on the i^{th} image, respectively. A high value of B_i indicates a high amount of non-text information in the image. An algorithm needs to eliminate all the non-text information to gain value equal to the benchmark.

The results of the single entry for this task from Yin et. al, evaluated using Wolf and Jolion method [18], are listed in Table 1. Yin et. al group the horizontal character candidates. Since a set of images contain curved text, and Indic scripts need a unique way of grouping, this algorithm entails a low recall value for the text localization task.

Figure 3 (a) shows the plot of benchmark values and algorithm result on each image in the dataset. Images with

Table 1: Performance of text locating algorithm (evaluated using Wolf and Jolion method) on the MASTER dataset. AS: algorithm strength (for normal and complex images).

Participant	Precision	Recall	F-score	AS (normal)	AS (complex)
Yin et. al	0.64	0.42	0.51	0.58	0.56

[3]Lluís Gómez and Dimosthenis Karatzas, Computer Vision Center, Universitat Autònoma de Barcelona, Spain.

[4]Ganesh K. Sethi, M. M. Modi College, Patiala, Punjab and Rajesh K. Bawa, Punjabi University, Patiala, Punjab.

Table 2: Performance evaluation of the algorithms submitted for the text segmentation task. Precision, recall and f-score values are given. Algorithm strength (AS) values are shown separately for normal and complex images.

Method/ Participant	Precision	Recall	F-score	AS (normal)	AS (complex)
Yin et. al	0.71	0.67	0.69	0.80	0.56
Gómez et. al	0.64	0.58	0.61	0.69	0.35
Sethi et. al	0.33	0.72	0.45	0.65	0.18
OTCYMIST	0.50	0.29	0.37	0.46	0.21

B_i below 20 are considered 'normal'; the others, 'complex'. AR_i values are close to B_i values for images with a large fraction of text pixels. As the fraction of text pixels reduces, the performance of the algorithm is erratic. The algorithm strength (AS) is estimated as,

$$AS_j = \sum_i AR_{ji} / \sum_i B_i \quad (3)$$

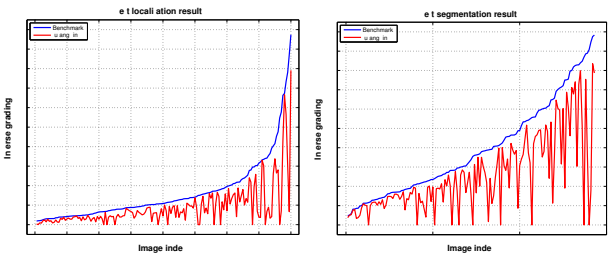
where, i is the index of range parameter for normal or complex images. The AS values for normal and complex cases are tabulated separately in Table 1.

5.2 Text segmentation task

We received three submissions for this task. Using OTCYMIST algorithm as the baseline, the submissions are evaluated and the results are given in Table 2.

Figure 3(b) shows the benchmark values and the results of the best algorithm for individual images in the data set. For normal images, the best algorithm has good results. However, as shown by Table 2, in the case of complex images, sometimes the result is either poor or bad, due to the degradations or the algorithm's post-processing threshold.

A moving average approach is used to include the result of all the algorithms in a single plot. A window of 11 images is moved through the images ordered by their B_i values. Mean values of B_i and AR_i are computed. Figure 4 shows the plot of these average values for text segmentation task. Top two algorithms use MSER algorithm during the



(a) (b)

Figure 3: A plot of benchmark values (B_i in blue colour) and algorithm result (AR_i in red colour) on the individual images in the MASTER data set for text (a) localization and (b) segmentation tasks. AR_i follows the B_i values in the case of normal images and fluctuates between high and low values in the case of complex images.

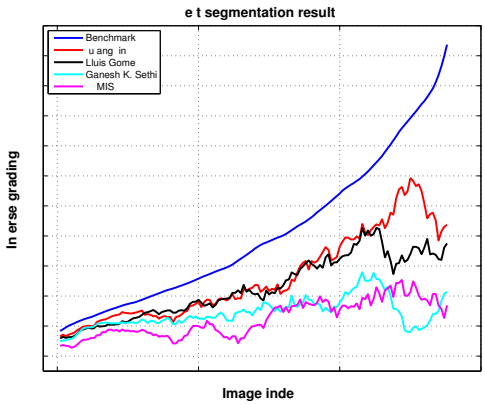


Figure 4: A plot of average values of B_i and AR_i for the algorithms submitted. The top performing algorithm is the nearest to follow the average values calculated from the benchmark.

character segmentation stage, and differ only at the post-processing stage. So, their segmentation results are poor on the images having illumination on text, text on glass or occluded text. The plots are close, revealing that the underlying methodology of the algorithms are similar. The results, where these two algorithms differ, are analyzed to figure out the reason. The top performing algorithm is more effective in removing non-text components, thereby increasing its precision. Neighboring components are removed while grouping and textured non-text components are not removed in Yin's method. If stroke width information of the characters is used, then all the components can be filtered out properly in the segmented image.

5.3 English word recognition task

At least three characters exist in each English word image. We did not receive any submission for this task. Our own algorithms are evaluated on the dataset for the purpose of benchmarking. Omnipage OCR [15] is used for recognition. The edit distance (ED) between the ground-truth and the output of the algorithm is calculated, giving equal weights to additions, deletions and insertions. ED of each word is normalized by the number of letters in the word and all the normalized EDs are accumulated to get the total edit distance (TED-E). The word recognition (EWR) rate and the TED-E are tabulated in Table 3 for different algorithms. Compared to ICDAR 2003 or 2011 word image datasets,

Table 3: Comparison of word recognition rate and total edit distance measures for English (EWR, TED-E) and Kannada (KWR, TED-K) for different algorithms, namely, Benchmark, PLT, MAPS, NESP and raw image.

Method	EWR	TED-E	KWR	TED-K
Benchmark [6]	57.7	215.1	11.1	178.7
PLT [10]	46.9	299.5	5.3	210.6
MAPS [7]	46.9	305.9	4.9	209.1
NESP [8]	45.1	305.4	5.8	212.3
Baseline (raw image)	37.5	369.4	2.5	218.5

an additional complexity of curved text is included in this MASTER dataset. One-third of the word images contain shear, slant or curved text. Thus, 33% of the words in the dataset cannot be directly recognized by standard OCR engines, since they handle only horizontal words. The number of words commonly recognized by all the three (PLT, NESP and MAPS) algorithms is 38.8% and the union of all the words recognized by these algorithms is 53.5%. These numbers indicate that a few words recognized by one algorithm are not recognized by others. The union result does not even cross the benchmarked recognition rate.

5.4 Kannada word recognition task

At least two Unicodes exist in each Kannada word image. We did not receive any submission for this task, and the dataset is benchmarked with our algorithms. Tesseract OCR engine [17] is used for recognizing Kannada word images. The ED is calculated at Unicode string level before normalization. The word recognition (KWR) rate and the total edit distance (TED-K) are given in Table 3 for the different algorithms. The words commonly recognized and the union of words recognized by all the three algorithms are 4.1% and 7%, respectively.

The recognition rate (RR) for manually segmented word images is low (both in English and Kannada) due to the additional complexity of curved or slanted text. If those are aligned before feeding them to OCR, then the benchmark RR may reach 90% for English words. The union of the recognized words is below the benchmark RR, indicating the need to improve word segmentation. In the case of Kannada words, part of the consonants appear in the descender region of a word. Care should be taken while segmenting these components for recognition, generating the Unicode and also while aligning the curved words.

6. CONCLUSION

A robust reading competition is organized on multi-script scene images. Each character in a script is confined within broadly defined three sections, namely ascender, x-height and descender. A set of script-related rules need to be defined to group characters and locate a word. Hence, the text localization is a more complex task than text segmentation. Yin et. al perform horizontal grouping, which works for Roman and Chinese, but not for Kannada and Devanagari.

A novel method is proposed to sort the images by inverse grading. The number of normal and complex images can be used as a measure to benchmark a dataset. The several object detection datasets available can be ranked based on the sum of benchmark values. Only the count of text or BB pixels is used to benchmark an image in this competition, which in turn benchmarks a dataset. Additionally, taking into account the pixels affected by degradation can further improve the benchmarking measure on the datasets.

Acknowledgment

The authors thank the MILE lab members, who provided scene images to be included in the database and others, who carefully annotated the scene images. The authors also thank Technology Development for Indian Languages (TDIL), DIT, for partially funding this research work.

7. REFERENCES

[1] Clavelli, A., Karatzas, D., and Llados, J. A framework for the assessment of text extraction algorithms on complex colour images. In *Proc. 9th DAS* (2010), pp. 19–28.

[2] ICDAR 2013. `http://www.icdar2013.org/`.

[3] Karatzas, D., Mestre, S. R., Mas, J., Nourbakhsh, F., and Roy, P. P. ICDAR 2011 robust reading competition - challenge 1: Reading text in born-digital images (web and email). In *Proc. 11th ICDAR* (2011), pp. 1485–1490.

[4] Kasar, T., Kumar, D., AnilPrasad, M. N., Girish, D., and Ramakrishnan, A. G. MAST: Multi-script annotation toolkit for scenic text. In *Proc. Workshop on J-MOCR-AND* (2011), pp. 1–8.

[5] Kasar, T., and Ramakrishnan, A. G. Multiscript and multioriented text localization from scene images. In *Proc. 4th CBDAR* (2011), pp. 15–20.

[6] Kumar, D., AnilPrasad, M. N., and Ramakrishnan, A. G. Benchmarking recognition results on camera captured word image datasets. In *Workshop on DAR* (2012), pp. 100–107.

[7] Kumar, D., AnilPrasad, M. N., and Ramakrishnan, A. G. MAPS: Midline analysis and propagation of segmentation. In *ICVGIP* (2012), p. 15.

[8] Kumar, D., AnilPrasad, M. N., and Ramakrishnan, A. G. NESP: Nonlinear enhancement and selection of plane for optimal segmentation and recognition of scene word images. In *DRR* (2013).

[9] Kumar, D., and Ramakrishnan, A. G. OTCYMIST: Otsu–Canny minimal spanning tree for born-digital images. In *Proc. 10th DAS* (2012), pp. 389–393.

[10] Kumar, D., and Ramakrishnan, A. G. Power-law transformation for enhanced recognition of born-digital word images. In *Proc. 9th SPCOM* (2012).

[11] Lee, S., Cho, M. S., Jung, K., and Kim, J. H. Scene text extraction with edge constraint and text collinearity link. In *Proc. 20th ICPR* (2010).

[12] Lucas, S. M. ICDAR 2003 robust reading competitions: entries, results, and future directions. *IJDAR 7*, 2-3 (2005), 105–122.

[13] Lucas, S. M. Text locating competition results. In *Proc. 8th ICDAR* (2005), pp. 80–85.

[14] MAST: Multi-script annotation toolkit for scene text. `http://mile.ee.iisc.ernet.in/mast/`.

[15] Nuance omnipage reader. `http://www.nuance.com/`.

[16] Shahab, A., Shafait, F., and Dengel, A. ICDAR 2011 robust reading competition challenge 2:reading text in scene images. In *Proc. 11th ICDAR* (2011), pp. 1491–1496.

[17] Tesseract OCR engine. `http://code.google.com/p/tesseract-ocr/`.

[18] Wolf, C., and Jolion, J. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR 8*, 4 (2006), 280–296.