ICDAR 2005 Text Locating Competition Results

Simon M. Lucas
Dept. of Computer Science
University of Essex, Colchester CO4 3SQ, UK
sml@essex.ac.uk

Abstract

This paper describes the results of the ICDAR 2005 competition for locating text in camera captured scenes. For this we used the same data as the ICDAR 2003 competition, which has been kept private until now. This allows a direct comparison with the 2003 entries. The main result is that the leading 2005 entry has improved significantly on the leading 2003 entry, with an increase in average f-score from 0.5 to 0.62, where the f-score is the same adapted information retrieval measure used for the 2003 competition.

The paper also discusses the web-based deployment and evaluation of text locating systems, and one of the leading entries has now been deployed in this way. This mode of usage could lead to more complete and more immediate knowledge of the strengths and weaknesses of each newly developed system.

1 Introduction

In recent years there has been some significant research into these general reading systems that are able to locate and/or read text in scene images [10, 2, 9]. As with all complex pattern recognition tasks, it is essential to quote results on standard datasets in order to have meaningful evaluation. The first publicly available ground-truthed dataset on which to evaluate such systems was that used for the ICDAR 2003 robust reading competitions. The test data for those competitions was kept private, and is used to assess the 2005 entries. The test dataset consists of 501 images captured with a variety of digital cameras. Cameras were used with a range of resolution and other settings, with and without flash, with the particular settings chosen at the discretion of the photographer. The images include household objects, road signs, shop signs, bill-boards and posters, and book covers. They span a wide range of apparent difficulties. A training dataset of 500 images of a broadly similar nature was made publicly available in Autumn 2002. Entrants were also free to tune their systems on their own data.

To enter the contests, researchers had to submit their software to us in the form of a ready-to-run command-line executable. This is known as *closed mode* evaluation. Closed mode evaluation has the advantage that once a system is submitted, it is not possible for its developers to perform any further tuning. Each entry takes a test-data input file and produces a raw results file. The raw results are then compared to the ground truth for that dataset by an evaluation algorithm, which produces a set of detailed results and also a summary. The detailed results report how well the algorithm worked on each image, while the summary results report the aggregate over all the images in the dataset. All these files are based on simple XML formats to allow maximum compatibility between between different versions of evaluation systems, recognizers and file formats.

Reading the text in an image is a complex problem that may be decomposed into several simpler ones. Competitions were to be run on text locating, character recognition, word recognition, and complete reading systems. Unfortunately, however, as for ICDAR 2003, the only competition to receive sufficient entries was the text locating contest. The lack of entries for the Robust OCR contest was a surprise, given that the data for this competition was made available in the popular MNist format, and normalised so that each image was within a 28×28 window. The datasets will remain on the web, and we encourage readers to use them for benchmarking.

2 Text Locating

The aim of the text locating competition was to find the system that can most accurately identify the word rectangles in an image. Note that several design options were possible here - such as specifying that the system find complete text blocks, or individual words or characters. We chose words since they were easier to tag and describe (it would be harder to fit rectangles to text blocks, since they are more complex shapes). The average number rectangles per image is 4.5, with a minimum of 0 and a maximum of 52.

The same evaluation scheme was used as for the ICDAR

2003 competition [5], based on the notions of precision and recall, as used by the information retrieval community. Precision, p is defined as the number of correct estimates divided by the total number of estimates. Systems that over-estimate the number of rectangles are punished with a low precision score. Recall, r is defined as the number of correct estimates divided by the total number of targets. Systems that under-estimate the number of rectangles are punished with a low recall score.

For text locating it is unrealistic to expect a system to agree exactly with the bounding rectangle for a word identified by a human tagger. Hence, we need to adopt a flexible notion of a match. We define the match m_p between two rectangles as the area of intersection divided by the area of the minimum bounding box containing both rectangles. This figure has the value one for identical rectangles and zero for rectangles that have no intersection. For each rectangle in the set of estimates we find the closest match in the set of targets, and vice versa.

Hence, the best match $m(r, R)$ for a rectangle r in a set of Rectangles R is defined as:

$$m(r, R) = \text{max } m_p(r, r') \mid r' \in R$$

Then, our new more forgiving definitions of precision and recall, where T and E are the sets of ground-truth and estimated rectangles respectively:

$$p' = \frac{\Sigma_{r_e \in E}\ m(r_e, T)}{|E|}$$

$$r' = \frac{\Sigma_{r_t \in T}\ m(r_t, E)}{|T|}$$

We adopt the standard f measure to combine the precision and recall figures into a single measure of quality. The relative weights of these are controlled by α, which we set to 0.5 to give equal weight to precision and recall:

$$f = \frac{1}{\alpha/p' + (1-\alpha)/r'}$$

2.1 Alternative Measures

A problem with the above measure is that it is rather difficult to interpret. As pointed out in [6], a recall of 0.9 could mean that all rectangles were found, each with an accuracy of 90%, or 90% of rectangles were perfectly identified, and the other 10% completely missed.

Wolf and Jolion in [6] propose an alternative more complex measure that allows a match quality criterion to be varied, with the result that for a specific quality we get an exact measure of the number of rectangles correctly found. Their measure goes further, in that it also caters for one-to-many and many-to-one matches. In [6], the results remain very similar under the new measure, however, so despite the limitations of our simple measure, this is the one we still use to rank the entries.

Another alternative form of evaluation would be a goal-directed approach [8]. In this case, the text locating algorithms could be judged by the word recognition rate they achieve when used in conjunction with a word recognizer (or OCR package). A difficulty of this approach, however, is its dependence on the particular recognizer used. A detailed evaluation of various object detection evaluation methods is given in [7].

3 The Submitted Systems

There were five entries for the 2005 competition. As in the case of the ICDAR 2003 competitions, many of the originally supplied entries were missing DLLs or other library files. Contestants were invited to supply any missing files, which they all did. A full description of the ICDAR 2003 text locating entries can be found in [6]. For the 2005 entries, we only have available at the time of writing descriptions of the two leading entries, which are given next.

3.1 The Hinnerk Becker System

This paragraph is adapted from a description provided by Hinnerk Becker:

> The system is “connected component” based, using an adaptive binarization method to extract character regions which are then combined to lines of text following some geometrical constraints. A number of (mostly edge or histogram based) features is calculated to classify these hypothesis as “text” or “non-text”. Then a horizontal and vertical vanishing point is estimated and the line is transformed into a frontal parallel view (although the vertical vanishing point for this contest as there is hardly any perspectively skewed text in the trial image set; instead it assumes weak perspective and estimates a common shearing angle.)

3.2 The Alex Chen System

The system was developed by Alex Chen and Alan Yuille. The following description was supplied by Alex Chen:

> The details of the text detection algorithm can be found in our previous publications [3], [4]. Our training set consists of both the ICDAR

2003 trial training dataset and our own urban images taken by blind and normally sighted subjects. From this dataset, we manually label and extract the text regions. Next we perform statistical analysis of the text regions to determine which image features are reliable indicators of text and have low entropy (i.e. feature response is similar for all text images). We obtain weak classifiers by using joint probabilities for feature responses on and off text. These weak classifiers are used as input to an AdaBoost machine learning algorithm to train a strong classifier. In practice, we trained a cascade with 10 stages. Each stage contains one strong classifier. Regions selected by the cascade classifier are clustered into groups according to their location and size. After that, an adaptive binarization algorithm is applied and connected components (CCPs) are extracted. Then the CCPs are grouped into lines followed by an extension algorithm to find missing boundary letters. Finally, we break the CCP lines into words for output.

4 Results

The text locating results on the private test data are shown in Table 1. The entries are identified by the user name of the person submitting each one. The column labelled $t(s)$ gives the average time in seconds to process each image for each system under test. All timings were made using a 2.4ghz PC running either Windows XP or Linux. Note that the *Full* system is the score obtained by returning a single rectangle for each image that covers the entire image. To give a baseline measure of processing time, this was computed by retrieving and decompressing each JPEG image using standard Java API methods, then measuring the image size.

The leading entry is by Hinnerk Becker. The second most accurate method is the 2005 submission by Alex Chen. Note that this is over 40 times faster than the winning entry, but less accurate. The *Jisoo Kim* entry crashed after processing the first 400 images (the results for *Jisoo Kim* are averaged over those images only). In future it would be much less effort if we could run these competitions by using an alternative mode of entry, where each competitor exposes their system as a web service. Progress has already been made in this direction, as discussed later in this paper.

Table 2 shows the number of times that each of the leading four algorithms scored the highest f value on an image, and also the number of times that all methods failed, with f close to zero. This shows that all the leading entries have something to contribute. Xiaofan Lin in [6] was able to significantly improve on the leading 2003 entries by combining their estimates with a specially developed combination scheme, and the same approach would be expected to work well here also, given that the systems make different mistakes.

System	precision	recall	f	t (s)
Hinnerk Becker	0.62	0.67	0.62	14.4
Alex Chen	0.60	0.60	0.58	0.35
Qiang Zhu	0.33	0.40	0.33	1.6
Jisoo Kim	0.22	0.28	0.22	2.2
Nobuo Ezaki	0.18	0.36	0.22	2.8
Ashida	0.55	0.46	0.50	8.7
HWDavid	0.44	0.46	0.45	0.3
Wolf	0.30	0.44	0.35	17.0
Todoran	0.19	0.18	0.18	0.3
Full	0.1	0.06	0.08	0.2

Table 1. Text locating results for the 2005 (top) and the 2003 (bottom) entries.

System	number of wins
Hinnerk Becker	242
Alex Chen	137
Ashida	78
HWDavid	44
Failed	14

Table 2. Number of times each of the top four systems scored highest on an image.

5 Visual Analysis of Results

We viewed many of the results of each program, especially the two leaders, to gain an impression of the strengths and weaknesses of each system. In each of the following images the ground truth rectangles are identified with long-dashed lines, and the estimates by short-dashed lines. The colors (grey-levels) have been chosen to be visible against the background. Each figure caption includes in parentheses the (p, r, f) scores for that image.

In many cases the performance of the leading two algorithms is impressive. Figure 1 shows the *Alex Chen* algorithm identifying all the words in a road sign, though also picking up some false rectangles in the building in the background, which leads to a low precision score.

Figure 2 shows the *Hinnerk Becker* system achieving a good result on an extremely blurred image, with the text on a CRT monitor, taken from a moving escalator.

Figure 3 shows a case where all algorithms miss the *Argos* shop sign. This is a case of highly stylized text that

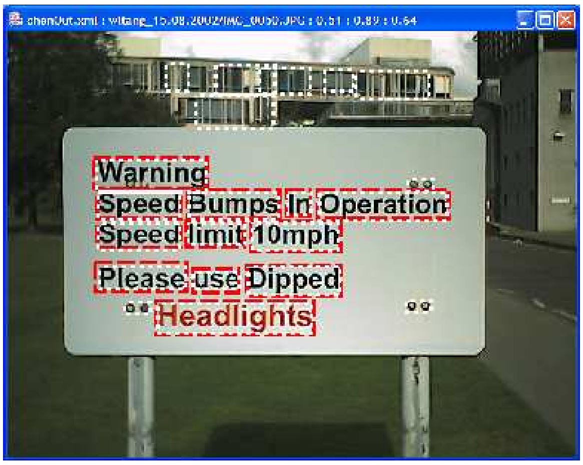

Figure 1. A road-sign image well handled by both leading algorithms; the shown estimates are for *Alex Chen* **(0.51, 0.89, 0.64).**

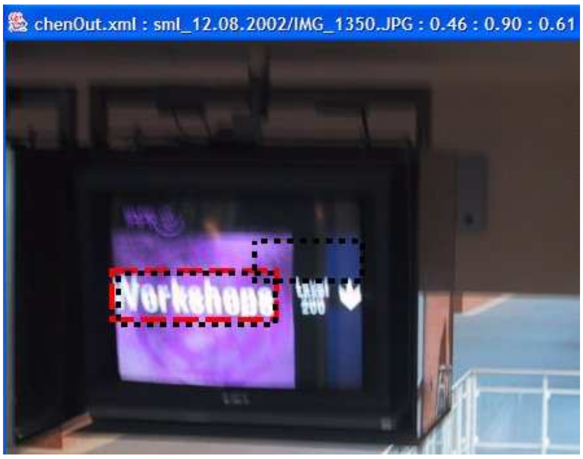
chenOut.xml : sml_12.08.2002/IMG_1350.JPG : 0.46 : 0.90 : 0.61

Figure 2. An with blurred text, but still located well by *Hinnerk Becker* **(0.46, 0.90, 0.61).**

people are able to locate and read, but confounds the algorithms under test.

6 Web Based Evaluation

Web-based deployment allows algorithms to be evaluated by end users or researchers, without the need to install the algorithm. This is a major advantage both for the end user, and for the algorithm developer. The end user is protected from lengthy installation procedures, which may also leave one's machine in a corrupted state. The algorithm developer is protected from potential theft of software or intellectual property.

In recent years there has been a great deal of interest in web services, and *Service Oriented Programming* [1] has been proposed as a new programming paradigm. Our system provides access to a deployed algorithm in two ways: interactive mode via a web browser, and program access mode via a special kind of web service architecture. Both deployment modes aim to offer immediate results, subject to server load. Web-browser mode is useful for users wishing to casually test systems on selected images. Program mode, on the other hand, can be used to test deployed algorithms on hundreds of test-cases, or even to build complete systems, where each component is a special kind of web service. The results of all text locating algorithm invocations are also logged to a web directory, allowing for subsequent public browsing and usage.

Figure 4 shows the results of uploading an image to the Text Locating service, using a web browser[1] (only a cropped version of the image is shown in order to provide reasonable resolution). Images uploaded in this way are not currently expected to have any associated ground-truth, though a future version of the service may provide for this possibility. Nonetheless, the immediate feedback provides a useful service, and users can quickly gauge the performance of an installed algorithm on the kind of images that matter to them. In addition to providing an image with the detected rectangles overlaid, the service also provides the rectangles marked up in XML.

7 Discussion and Conclusions

Running the text locating contest provides some insights into the strengths and weaknesses of the submitted systems. These can be summarized as follows:

- Most easy-to-read (for humans) text is now well detected.
- There was a major difference in the speed of the submitted systems, with Alex Chen being over forty times faster than Hinnerk Becker.
- Variations in illumination, such as reflections from light sources cause significant problems.

If these results are indicative of the state of the art in text locating, then there has been a significant advance in performance in the two years between the 2003 and the 2005 competitions, with a better than 10% increase in the average f measure of the respective winning entries. By another measure, the top two 2005 entries out-perform the top two 2003 entries on 76% of the test images (the ratio is very similar if we compare just the leading entries).

[1] http://algoval.essex.ac.uk:8080/textloc/upload.html

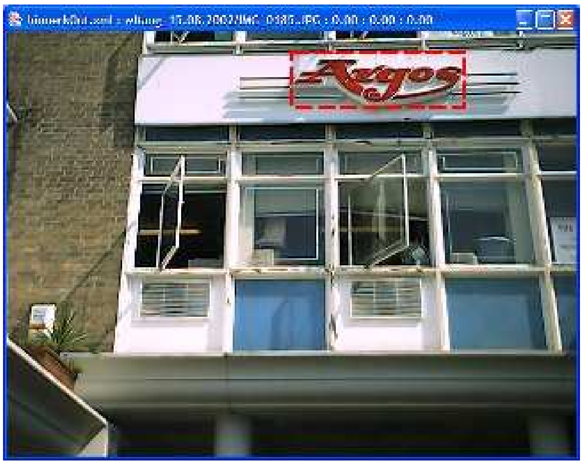

Figure 3. An image where all algorithms miss the text (0.0, 0.0, 0.0).

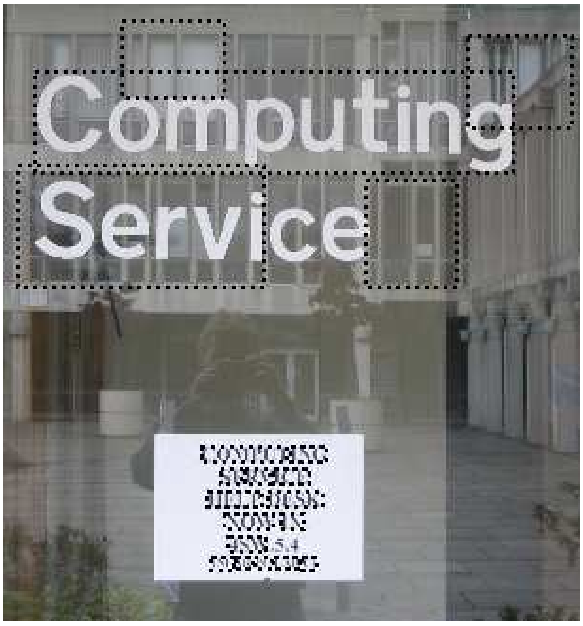

Figure 4. An image uploaded to the Alex Chen algorithm, installed as a web service. The located rectangles are shown in black dotted lines.

Worth noting is the option for web-based deployment of text-locating systems. The system by Alex Chen is already deployed in this way, and can be tested by users on novel images uploaded using a web browser. It would be great to see this kind of deployment become the norm, allowing for easier evaluation of new systems.

Text locating algorithms are clearly improving, at least for the way the problem has been specified for these competitions. The hope now is to also make progress on the other problems related to reading text in scenes.

Acknowledgements Thanks go to Alex Chen for allowing his text locater to be deployed as a web service.

References

[1] G. Bieber and J. Carpenter. Introduction to service-oriented programming (rev 2.1). `http://www.openwings.org/download/specs/ServiceOrientedIntroduction.pdf` .

[2] J. Chang, X. Chen, A. Hanneman, J. Yang, and A. Waibel. A robust approach for recognition of text embedded in natural scenes. *Proceedings of International Conference on Pattern Recognition*, pages 204 – 207, 2002.

[3] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:366 – II:373, 2004.

[4] X. Chen and A. L. Yuille. A time efficient cascade for real-time object detection: with applications for the visually impaired. In *Proceedings of the CVAVI05, IEEE Conference on Computer Vision and Pattern Recognition Workshop*, page to appear, 2005.

[5] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, pages 682 – 687. IEEE Computer Society, 2003.

[6] S. M. Lucas and et al. Icdar 2003 robust reading competitions: Entries, results, and future directions. *International Journal of Document Analysis and Recognition*, page to appear, 2005.

[7] V. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, H. Li, D. Doermann, and T. Drayer. Performance evaluation of object detection algorithms. In *Proceedings. 16th International Conference on Pattern Recognition, Volume 3*, pages 965 – 969. IEEE Computer Society, 2002.

[8] O. Trier and A. Jain. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:1191–1201, 1995.

[9] C. Wolf, J.-M. Jolion, and F. Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 1037–1040, 2002.

[10] V. Wu, R. Manmatha, and E. M. Riseman. Finding text in images. In *Proceedings of 2nd ACM Conference on Digital Libraries*, pages 3–12, 1997.