Recursive Text Segmentation for Color Images for Indonesian Automated Document Reader

Teresa Vania Tjahja*[1], Anto Satriyo Nugroho#[2], Nur Aziza Azis#,
Rose Maulidiyatul Hikmah#, James Purnama*
*Faculty of Information Technology, Swiss German University
EduTown BSD City, Tangerang 15339, Indonesia
#Center for Information and Communication of Technology (PTIK),
Agency for the Assessment and Application of Technology (BPPT)
Jalan M.H. Thamrin No. 8, Jakarta 10340, Indonesia
Email: [1]teresa.vania@gmail.com, [2]asnugroho@ieee.org

***Abstract*—Indonesian Automated Document Reader (I-ADR) is an assistive system for Indonesian citizens with visual impairment, which converts textual information on papers to corresponding speech. I-ADR system is designed to be operated via a voice-based user interface. The system accepts document images as inputs and employs Optical Character Recognition (OCR) and Text-to-Speech (TTS) Synthesizer technology to read the image. This research is focused on Text Segmentation module as an integral part of OCR module, both for color and grayscale images. The Text Segmentation module implements Multivalued Image Decomposition algorithm[1] and Enhanced Constrained Run-Length Algorithm[2], equipped with our proposed recursive method. During the experiments, the algorithm achieved 96% success rate.**

I. Introduction

As technology advances, many documents that were originally presented on papers are converted into their electronic form. These electronic documents, complemented with assistive software such as screen readers, improve the accessibility of visually-impaired people to textual information on various documents. However, in developing countries such as Indonesia, paper is still the most common medium for carrying information. Paper documents can be made available electronically, for example by scanning or taking photograph of the document, but most of the devices produce images of the documents, in which the text could not be read directly by screen readers. Numerous systems have been developed to convert textual information from document images to speech, but most of them are designed to read documents in foreign languages.

Indonesian Automated Document Reader (I-ADR) is an assistive system developed by the Agency for the Assessment and Application of Technology *(Badan Pengkajian dan Penerapan Teknologi; BPPT)* to help citizens with visual impairment to obtain textual information on paper documents. The system uses Optical Character Recognition (OCR) technology involving text segmentation and character recognition (conversion from character image to machine-encoded text), and also performs word correction tailored with Indonesian dictionary words for more accurate results. In the past few years, several studies[3], [4] have produced a prototype of I-ADR, resulting in integration of OCR, Text Summarization, and TTS Synthesizer modules. However, the prototype was designed to read only from grayscale images.

In this research, we evaluate the text segmentation algorithm that has been implemented for grayscale images and enhance the algorithm for use with color images. The remainder of this paper is organized as follows: the implementation of the proposed I-ADR system is presented in Section II with the main focus on OCR module, experimental results are discussed in Section III, and will be concluded in Section IV.

II. Proposed System

In general, Indonesian Automated Document Reader consists of 4 main modules, as shown in Figure 1: voice-based user interface, Optical Character Recognition (OCR), Text Summarization, and Text-to-Speech (TTS) Synthesizer. A typical use case is when a user commands the system to read a document image. In this scenario, the document image is given as an input to the OCR module. The OCR module then performs text segmentation and character extraction to acquire textual information from the image. The resulting machine-encoded text is then passed to TTS Synthesizer, which converts the text into speech. As an alternative, the user can also choose to listen to the summary of textual information on the document. In this case, the result from OCR module is passed to Text Summarization module,which in turn extracts the summary. Then, TTS Synthesizer converts the summary into speech. This information flow is depicted in Fig. 2.

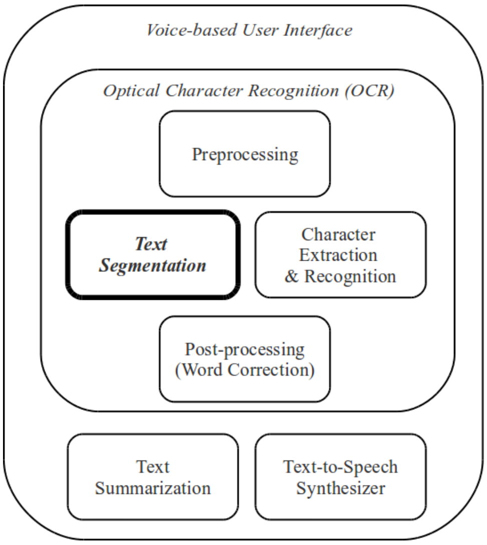

Voice-based User Interface
Optical Character Recognition (OCR)
Preprocessing
Text Segmentation
Character Extraction & Recognition
Post-processing (Word Correction)
Text Summarization
Text-to-Speech Synthesizer

Fig. 1: Indonesian Automated Document Reader modules.

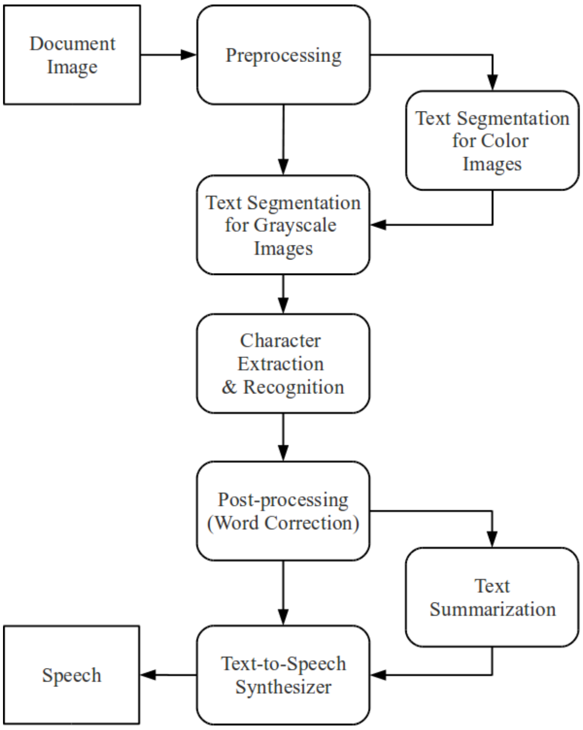

Document Image
Preprocessing
Text Segmentation for Color Images
Text Segmentation for Grayscale Images
Character Extraction & Recognition
Post-processing (Word Correction)
Text Summarization
Speech
Text-to-Speech Synthesizer

Fig. 2: Flowchart of relationships between I-ADR modules.

In the following subsections, the OCR module will be discussed in more details.

A. Pre-processing

Pre-processing module prepares the input image for subsequent image processing operations. To enhance image quality, median filter[5] is used to remove noise from the image and binarization using Otsu's thresholding method[5] is utilized to simplify input image representation. However, these methods are applied only for grayscale images. Processing color images will be discussed in the next subsection.

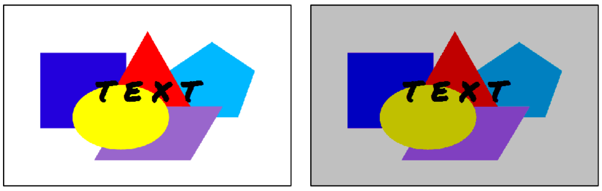
B. Text Segmentation

In the existing I-ADR prototype, the Text Segmentation module implements Enhanced Constrained Run-Length Algorithm[2] equipped with our proposed recursive method[4]. However, the algorithm takes binary images, which has only two intensities (foreground and background) for its pixels, as inputs. For a color image to be processed properly by the algorithm, it needs to be simplified into binary images. It is then achieved by applying Multivalued Image Decomposition algorithm[1] on the color images.

The idea in Multivalued Image Decomposition algorithm is to reduce the number of colors used in an image with bit-dropping and color quantization. In bit-dropping, only two most significant bits from each color channel (Red, Green, and Blue) are used, while the remaining bits eliminated/dropped. Then, color quantization merges colors through single-link clustering. The image with reduced color is then decomposed into several images based on colors. For instance, one image of decomposition result would contain objects with the same color from the input image. Fig. 3 shows the example of a color image with its colors reduced. The decomposition result is given in Fig. 4.

It can be seen from Fig. 4 that each image produced by the decomposition process has only two intensities: a color used in the input image as the foreground color, and a background color (usually set to white). Even for image that contains background objects from the input image, there are only two intensities with the background objects treated as foreground objects in the decomposition result. Thus, the decomposition results can be considered as binary images. Text segmentation and character extraction from these images can then be performed with the algorithms that have also been used for grayscale images.

As stated previously, Text Segmentation module of I-ADR utilizes Enhanced Constrained Run-Length Algorithm (CRLA)[2]. Basically, Enhanced CRLA groups objects in the input image based on their size, so that neighboring objects with similar sizes are grouped into one homogeneous region. A sample result of applying Enhanced CRLA to a document image is shown in Fig. 5.



(a) Original color image. (b) Image 3a with reduced colors.

Fig. 3: Example result of color reduction.

Each of the homogeneous regions in Fig. 5 is then classified as text or non-text based on their Mean Black Run-Length (*MBRL*) and Mean Transition Count (*MTC*) values [2]. A region is classified as text if its *MBRL* and *MTC* values lie between the pre-defined threshold values. Given this scheme, the main challenge is to segment the regions, so the *MBRL* and *MTC* values are calculated for an individual region. If the calculation is performed on each individual region, the obtained values are assured to actually represent the characteristics of the region. Otherwise, if a candidate region (the one being examined) consists of several regions of different types (i.e. not all text), the non-text regions may alter the *MBRL* and *MTC* values to fall outside the range for text. As a result, text regions included in the candidate region would not be extracted.

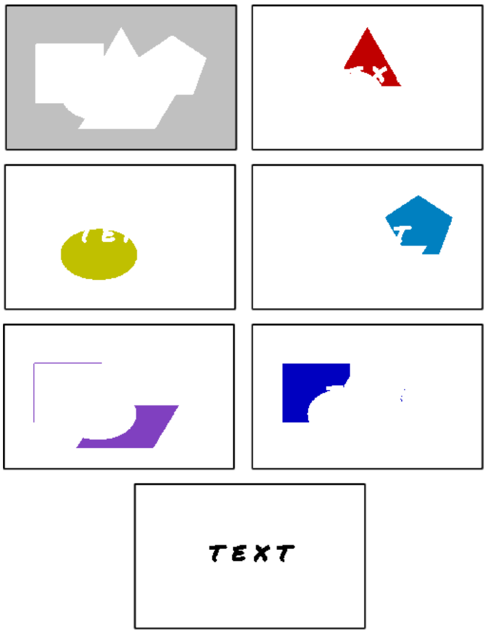
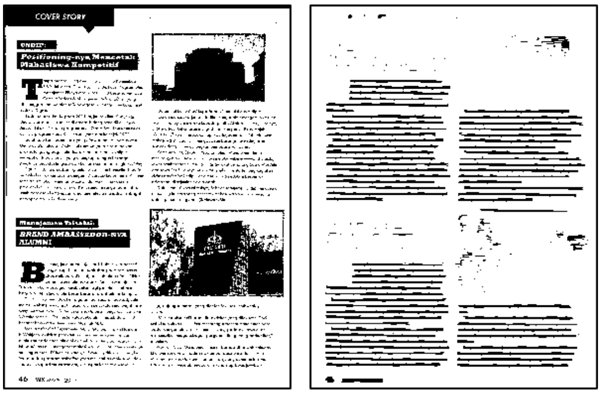


Fig. 4: Decomposition results of the original image in Fig. 3a.



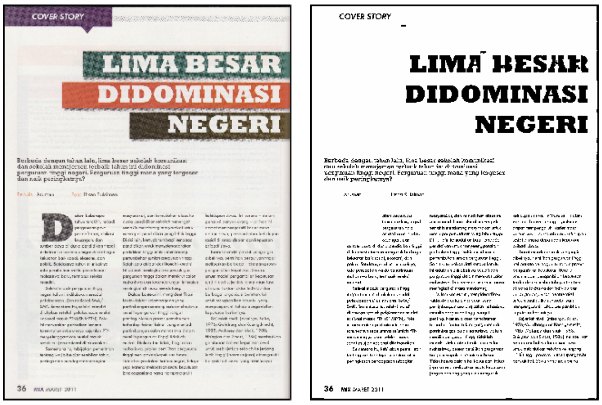
(a) Original binary image.

(b) Result after applying Enhanced CRLA.

Fig. 5: Homogeneous regions formed by Enhanced CRLA. Each black areas (text lines or parts of pictures) are considered as one individual region.

```
DETECT-TEXT-REGIONS(IMG, Xstart, Xend, Ystart, Yend)
Perform vertical scan
    Find starting and ending row (ys and ye) of
      a candidate region
    In the range of starting and ending row,
      perform horizontal scan
        Find starting and ending column (xs and xe) of
          a candidate region
        In the range of starting point (xs, ys) to (xe, ye),
          perform vertical scan
            Find starting and ending row (y2s and y2e)
            if xs ≠ Xstart AND xe ≠ Xend
              AND y2s ≠ Ystart AND y2e ≠ Yend
                DETECT-TEXT-REGIONS(IMG, xs, xe, y2s, y2e)
            else
                Mark region (from (xs, ys) to (xe, ye)) as
                  text or non-text based on its MBRL and MTC
```

Fig. 6: Pseudocode of the proposed recursive method.



(a) Original document image.

(b) Final text segmentation result.

Fig. 7: Result of the proposed text segmentation algorithm.

Our proposed recursive method is developed to overcome this problem. The idea is to scan through the image, while detecting the location of foreground and background pixels along the way, in order to get starting and ending coordinates of each region[4]. A pseudocode briefly explaining this recursive method is presented in Fig. 6, while the detailed one can be found in [4].

C. Character Extraction and Recognition

Text Segmentation module extracts text from input document image and produces a binary image containing the extracted text as shown in Fig. 7. This result is then passed to Character Extraction and Recognition module, which is responsible for extracting individual characters and converting those characters from image to machine-encoded (ASCII) text.

Character extraction (segmentation) is performed also with histogram analysis. The segmented character image is then normalized into 12 x 12 pixels and recognized using Multilayer Perceptron (MLP) neural network classifier. The MLP neural network has 3-layer structure consisting of input, hidden, and output layer, with each layer having 144, 100, and 73 nodes, respectively. The terminating condition for MLP neural network training phase is when it has performed 10,000 iterations or the Mean Square Error (MSE) is less than or equal 0.0001. In addition to the neural network classifier, symbols such as period (.), comma (,), and single/double quotes are recognized with rules, which are based on the character size and position in the text line.

At this point, a string of recognized characters is obtained. To produce an understandable text, the characters must be arranged into words by inserting spaces between the characters. The position of a space between two words is approximated as 2.6 times (obtained experimentally) of the average horizontal distance between characters in a word. Thus, for each detected character through histogram analysis, the distance between it and the previous character is calculated. When the distance is larger than 2.6 times the average distance in the word, a space is inserted.

D. *Post-processing*

In character recognition process, a character may be recognized as another with similar shape. For example, I is recognized as l or the number one (1), o as the number 0, etc. Therefore, to produce meaningful speech, words produced by character recognition module should be corrected before passed to the Text-to-Speech Synthesizer. I-ADR implements a lexicon-based post-processing to perform word correction. The algorithm compares recognized words with a list of Indonesian dictionary words. To measure the similarity between two words, Longest Common Subsequence (LCS)[6] is used.

III. Experimental Results and Discussion

Indonesian Automated Document Reader was developed and evaluated on a notebook PC with Intel Core i7 CPU @ 1.7 GHz and 4 GB RAM running Ubuntu Linux 10.10. The system has been tested with 14 document images, 7 grayscale and 7 color images. Those test data were acquired by scanning Indonesian magazine pages (A4 size) with 300 dpi scanner in clean condition and minimum skew. The average size of the images is 2381 x 3153 pixels. Originally, images obtained during the scanning process were in colors. To obtain grayscale images, the color images were converted to grayscale with *convert* utility in Ubuntu Linux 10.10.

The experimental results and discussion is presented in the following subsections, with emphasis on text segmentation experiments.

A. *Text Segmentation Experiments*

Text Segmentation module was evaluated with grayscale and color images as stated previously. Initially, experiments with grayscale images were conducted with the following conditions: 1) the proposed recursive method was not used, 2) the recursive method was used. Table I shows text segmentation result without recursive method, while Table II shows the result with recursive method.

TABLE I: Text segmentation result without recursion.

		Text Lines			
No.	Image	Total	Extracted	Acc.	Time
				(%)	(s)
1	magz 01	84	84	100	7.2
2	magz 02	90	90	100	10.1
3	magz 03	140	140	100	11.2
4	magz 04	115	115	100	10.4
5	magz 05	84	84	100	10.4
6	magz 06	67	56	88	10.6
7	magz 07	115	115	100	10.1
			Average	98	10

TABLE II: Text segmentation result with recursion.

		Text Lines			
No.	Image	Total	Extracted	Acc.	Time
				(%)	(s)
1	magz 01	84	84	100	7.2
2	magz 02	90	90	100	10.3
3	magz 03	140	140	100	11.2
4	magz 04	115	115	100	10.6
5	magz 05	84	84	100	10.9
6	magz 06	67	67	100	10.5
7	magz 07	115	115	100	10
			Average	100	10.1

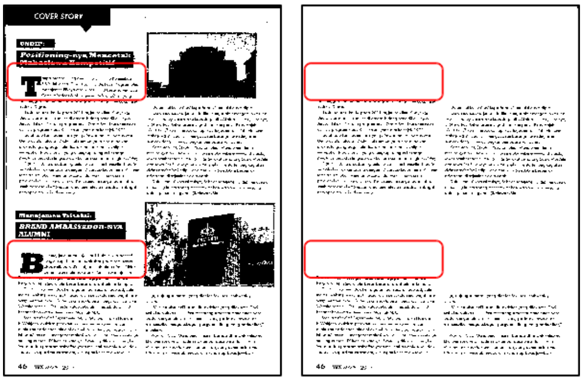


Fig. 8: Text segmentation result of image *magz 06*.

As can be seen in Table I, there are missing text lines in image *magz 06* when the algorithm was implemented without recursion. This problem is shown in Fig. 8. The missing text lines were located on an area with complex layout, which are marked as red in the images (see [4] for more detailed explanation). After the recursive method was implemented, all text lines can be extracted from all images, as shown in Table II. The column Accuracy (Acc.) in both tables indicates the ratio between the extracted/segmented text lines over the total expected text lines.

Meanwhile, Table III shows text segmentation result for color images. There are differences between total text lines in color and grayscale images experiments, because the algorithm for grayscale images was expected to extract text that is colored black as foreground and white as background, while the algorithm for color images was expected to extract text in any color.

It can be observed from Table III that there are text

TABLE III: Text segmentation result with color images.

No.	Image	Text Lines Total	 Extracted	Acc. (%)	Time (s)
1	magz 01	87	86	99	17.1
2	magz 02	90	90	100	29.6
3	magz 03	158	145	92	30.1
4	magz 04	115	115	100	23.4
5	magz 05	87	80	92	24.5
6	magz 06	74	68	92	27.3
7	magz 07	115	115	100	20.3
			Average	96	24.6

lines that could not be extracted in some images. Upon further examination, these missing text lines were not extracted because of *MBRL* and *MTC* thresholds used in Enhanced CRLA - the missing lines have *MBRL* and *MTC* values outside the threshold ranges, and thus considered by the algorithm as non-text.

Particularly in image *magz 03*, the missing text lines were caused by noise in the decomposition result. Fig. 9 shows magz 03 original image and its result after color reduction process. That result was decomposed into several images. One of the decomposition result is displayed in Fig. 10, along with its resulting image after Enhanced CRLA was applied on that decomposition result. As can be seen in Fig. 10, text in the table (at the center-top area of the image) is surrounded by noise due to the color used in the original color image. Consequently, those texts and the surrounding noise were grouped into homogeneous regions. In turn, this noise altered *MBRL* and *MTC* values of those homogeneous regions, causing the regions to be considered as non-text and thus was not extracted. The final text segmentation result is presented in Fig. 11.

B. Character Recognition Experiments

So far, character recognition in I-ADR system is implemented using MLP neural network classifier, with several symbols and punctuation marks recognized with rules based on the character size and position. Only results from 5 test images, whose layouts represent the other images, were reported. The recognition result is 98.31% for letters and numbers, and 94.71% for symbols and punctuation marks (shown in Table IV and V). For letters and numbers, the 1.69% error rate does not include errors that are caused by similar character shapes, such as 1 as l or I, and o as O or 0.

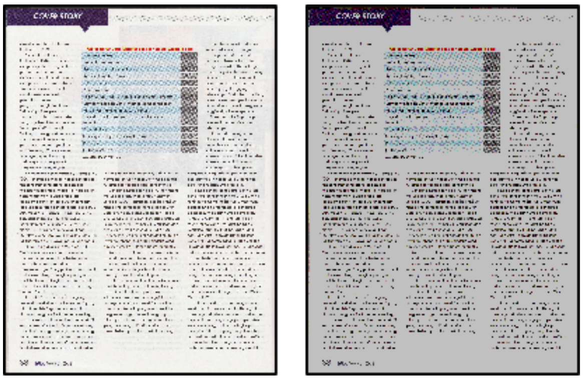


Fig. 9: Original image (left) and result after color reduction (right) of image *magz 03*.

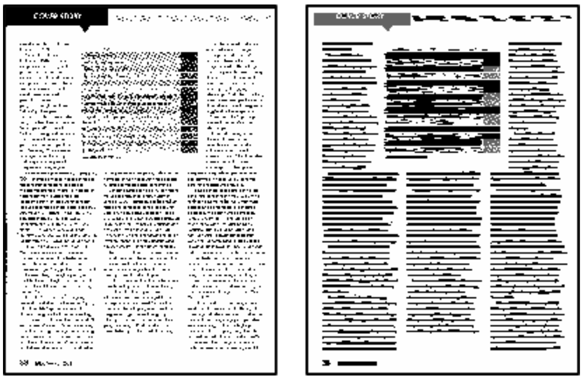


Fig. 10: One of the decomposition result of image *magz 03* (left) and its result after applying Enhanced CRLA (right).

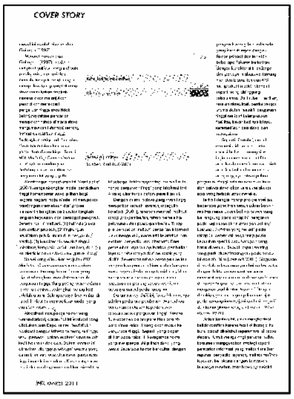


Fig. 11: Final text segmentation result of image *magz 03*.

C. Post-processing Experiments

Character Extraction and Recognition module takes images as the inputs and produces a text file containing strings of recognized characters as the output. The next module in the system, the Post-processing module, reads the text file and attempts to correct each word (string of characters) by comparing the recognized word with a list of dictionary words. For experimental purpose, the dictionary used in this module consists of

TABLE IV: Character recognition result for letters and numbers.

No.	Image	Total Char.	Errors Quantity	 %	Recognition Rate (%)
1	magz 01	2434	30	1.23	98.77
2	magz 02	2955	30	1.02	98.98
3	magz 04	3239	63	1.95	98.05
4	magz 05	3529	79	2.24	97.76
5	magz 06	3209	65	2.03	97.97
				Average	98.31

TABLE V: Character recognition result for symbols and punctuation marks.

			Errors		
No.	Image	Total Sym.	Quantity	%	Recognition Rate (%)
1	magz 01	60	9	15.0	85
2	magz 02	86	4	4.65	95
3	magz 04	107	4	3.74	96
4	magz 05	116	2	1.72	98
5	magz 06	74	1	1.35	99
				Average	95

TABLE VI: Word correction result.

			Errors		
No.	Image	Total Words	Quantity	%	Correction Rate (%)
1	magz 01	358	18	5.03	95
2	magz 02	446	14	3.14	97
3	magz 04	584	31	5.31	95
4	magz 05	559	41	7.33	93
5	magz 06	503	32	6.36	94
				Average	95

words obtained from the magazine pages used as test data.

The results of Post-processing module are given in Table VI, which shows 94.57% accuracy (correction rate). Due to bad segmentation result during character extraction process, there are words that could not be corrected perfectly, which might happen because there were too many character recognition errors in that word. Furthermore, a word might be broken into several strings of characters because the spaces were inserted in the wrong position. Since a word in Post-processing module is defined as *a string of characters that are enclosed by two spaces* (one on the left and one on the right), space insertion errors may cause a word to be divided into several words or several words combined into one.

IV. Conclusion

Indonesian Automated Document Reader is a project of the Agency for the Assessment and Application of Technology, which is intended to help Indonesian citizens with visual impairment to obtain textual information from paper medium. I-ADR system has the capability of reading document images and converting the textual information to speech. Given the main functionality of the system, which is focused on the textual information, the main object of interest in an image is the text itself. Extraction of textual information on an image requires a series of steps, including text segmentation and Optical Character Recognition (OCR). Those steps convert text in an image to machine-encoded text, which is required by the Text-to-Speech (TTS) Synthesizer as the module that generates the output speech. This study aims to develop and evaluate text segmentation algorithms for I-ADR system.

In this research, text segmentation algorithms for both grayscale and color images have been developed and evaluated with test data consisting of scanned magazine pages with complex layouts. Text segmentation for grayscale images utilizes Enhanced Constrained Run-Length Algorithm [2], which is equipped with our proposed recursive method to further increase its accuracy for extracting text areas. The proposed algorithm for grayscale images also serves as the basis of text segmentation algorithm for color images. Since the implemented Enhanced CRLA accepts grayscale (or binary) images as input, the representation of a color image should be simplified to that of a grayscale image before it can be processed with the algorithm. However, this simplification is not as straight as converting the colors themselves, but important information (especially the one concerned with the existence of text) must be extracted and presented in the produced grayscale image. To ensure that important information is extracted while simplifying the image representation at the same time, Multivalued Image Decomposition [1] is used. This algorithm decomposes a color image to several binary images. Then, detection of text area in each of the binary images is done by applying Enhanced CRLA to the image.

Despite the relatively high success rates, the proposed text segmentation algorithms are still far from perfect. Besides problems that are discussed in the previous section, there are many cases that have not been anticipated in the current implementation of Text Segmentation module. Thus, the Text Segmentation module needs enhancements, including skew correction, algorithm improvement for pages with non-Manhattan layouts, and processing degraded documents.

In broader view, the current prototype of I-ADR should be developed further in order to truly achieve its objective to be an assistive system for people with visual impairment. The future works of I-ADR include integration of voice-based user interface, improvement of Text-to-Speech Synthesizer module to produce speech with intonation, and system evaluation with large-scale data.

References

[1] A. K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. In *Proc. of 14th International Conference on Pattern Recognition*, pages 1497–1499, 1998.

[2] H. M. Sun. Enhanced Constrained Run-Length Algorithm for Complex Layout Document Processing. *International Journal of Applied Science and Engineering*, 2006.

[3] A. S. Nugroho, I. Suwadi, V. Pragesjvara, R. Irbandini, O. Riandi, M. Gunawan, and D. Handoko. TextReader: An Intelligent System for People with Visual Impairment. In *Proc. of International Conference on Advance Computer Science and Information System 2010*, pages 393–396, 2010.

[4] T. V. Tjahja, A. S. Nugroho, J. Purnama, N. A. Azis, R. M. Hikmah, O. Riandi, and B. Prasetyo. Recursive Text Segmentation for Indonesian Automated Document Reader for People with Visual Impairment. In *Proc. of International Conference on Electrical Engineering and Informatics 2011*, 2011.

[5] R. C. Gonzales and R. Woods. *Digital Image Processing*. Pearson Education Inc., 3rd edition, 2010.

[6] N. C. Jones and P. A. Pevzner. *An Introduction to Bioinformatics Algorithms*. The MIT Press, 2004.