A Model for Automatic Recognition of Vertical Texts in Natural Scene Images

Ong Yi Ling, Lau Bee Theng, Almon Chai
Faculty of Engineering, Computing, and Science
Swinburne University of Technology Sarawak Campus
Sarawak, Malaysia
{YLOng, blau, achai}@swinburne.edu.my

Chris McCarthy
Faculty of Science, Engineering, and Technology
Swinburne University of Technology
Melbourne, Australia
cdmccarthy@swin.edu.au

***Abstract*—Text recognition plays an important role in recognizing texts presented in the images as they provide important information. Scene text recognition has been an active research topic with rapid growth of development to improve the performance of text recognition with better reliability and accuracy. However, scene text recognition is challenging due to images containing inconsistent lighting, low resolution and blurriness. In addition, scene texts are usually taken from outdoor signboards, signage and road signs, which contain various orientation and fancy font styles to attract attention. Various researchers have proposed methods for recognizing different orientations of scene texts, such as horizontal texts, curved texts and rotated texts. However, to data there is a lack of research in recognizing vertical texts in natural scene images. In this research, a model for effective automatic recognition of vertical texts in natural scene images has been proposed, consisting of two major processes which are text localization and segmentation and text recognition. This proposed model recognizes three different types of vertical scene texts, which are top-to-bottom vertical texts, bottom-to-top vertical texts and horizontal-stacked vertical texts.**

***Index Terms*—Scene text recognition, vertical texts, scene image, segmentation, connected component, text region, Optical Character Recognition.**

I. Introduction

Text is an important medium for conveying information. Therefore, text recognition is important in accessing this information, particularly for assisting people with visual impairment [1]. For example, text recognition may assist independent travel, such as getting directions, identifying objects or even searching for products in the supermarket.

Images that contain text can be differentiated into two categories: scanned documents and scene images with text [2]. Scanned documents normally contain clean and uniform background, while scene images have complicated backgrounds with other objects. Hence, text recognition for scene images is more challenging as scene texts consist of various colors, resolutions, font sizes, complex background and lighting conditions [3]. For example, some objects in the scene appear to be similar with text, such as the wheels of a vehicle which may be detected as the letter 'O' [4].

II. Related Studies

In general, the common process of a scene text recognition is shown in Fig. 1. Starting with an input image, the image undergoes text detection which involves text localization and verification [5]. Text localization attempts to localize text regions in the image, while verification is to determine the false positives detection. After that, text segmentation and recognition is required to filter out the false positives detection and finally converts the text detected in the image into strings that can be understood by the computer.

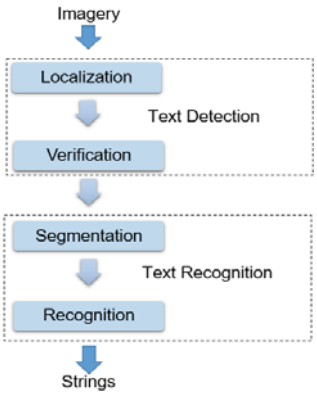


Fig. 1. Stepwise method for text detection and recognition.

Fig. 2. (a) Images containing horizontal scene text [6], (b) rotated scene text [7], (c) curved scene text [8] and (d) vertical scene text.

(a) (b) (c)

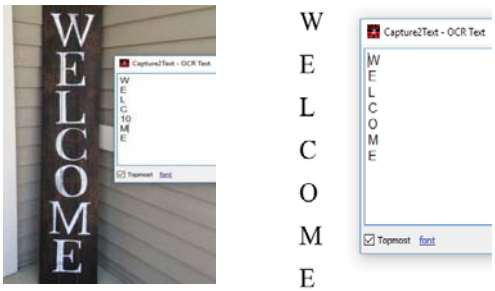
Fig. 3. (a) Different types of vertical texts such as horizontal-stacked vertical texts, (b) bottom-to-top vertical texts and (c) top-to-bottom vertical texts.

Besides that, scene text tends to appear in different orientations, such as horizontal texts, rotated texts, curved texts and vertical texts as shown in Fig. 2. Text recognition has become a popular research topic in recent years. Many researchers have proposed various approaches for text recognition system in recognizing different text orientations, such as horizontal texts, curved texts and rotated texts as mentioned in [5], [9] and [10].

[3] has presented a multi-oriented text detection using end-to-end fully convolutional network. [7] also proposed text detection for multi-orientation that perform classification and regression using rotation sensitive and insensitive features. Using image partition and connected components grouping, [11] performs text detection from text characters to text strings. Various approaches have been proposed for text recognition system, but they have limitation on detecting vertical scene texts.

Limited research has been done on vertical scene text recognition. Vertical scene text can be differentiated into horizontal-stacked vertical texts, top-to-bottom vertical texts and bottom-to-top vertical texts as shown in Fig. 3. It is important in recognizing these different types of vertical texts as they could provide useful and additional source of information to understand the natural scenes. For example, vertical scene texts could show shop names, warning signs, advertisements, shelves labelling and more.

Capture2Text [12] presented a model that is able to run OCR based on a text region selected manually by the user. It is developed to recognize horizontal texts in images containing Japanese manga comics. As the letters in horizontal-stacked vertical texts appears as horizontal letters, it is able to detect the letters in horizontal-stacked vertical texts, but it is unable to read the detected letters as a single word. Besides that, it is unable to detect bottom-to-top vertical texts and top-to-bottom vertical texts. Capture2Text [12] does not detect texts autonomously as it requires user to select and identify the text area manually. When an image with scene texts is load as an input image, the user has to select the text area manually and eliminate the background by selecting the text area only, so that the model will not detect non-text regions as text candidates which affects the text recognition. Fig. 4 shows Capture2Text [12] detects well with clean background but not for scene images as there are some false detections occur.



(a) (b)

Fig. 4. Detection obtained using Capture2Text [12].

Apart from detecting horizontal scene texts, rotated scene texts and curved scene texts, there is a lack of research in recognizing vertical texts in natural scene images as to date. As vertical texts could provide additional and useful source of information to understand the natural scenes, it is necessary to develop a model for recognizing vertical texts in natural scene images autonomously.

III. Proposed Model

Various models have been established for recognizing horizontal texts, rotated texts and curved texts in natural scene images. However, there is a lack of recognition in vertical texts as compared to the success of recognition texts in other orientation. As mentioned in [3], the proposed method is unable to detect vertical scene texts. In this research, a model for detecting and recognizing vertical scene texts in natural scene images is developed. The objective of this research is to design an effective automatic scene text recognition model for detecting and recognizing top-to-bottom vertical texts, bottom-to-top vertical texts and horizontal-stacked vertical texts in natural scene images. There are two major processes in this proposed model: text localization and segmentation, and text recognition.

A. Text Localization and Segmentation

The process of text localization and segmentation begins with an input scene image with vertical text. In this process, the input image undergoes grayscaling, followed by Maximally Stable Extremal Regions detector, which is known as MSER, to determine the possible candidate characters [1]. After that, binarization and dilation take place to obtain a binary image showing possible candidate characters. Lastly, connected component segmentation takes place to eliminate false positives [8].

Grayscaling is a process that uses gray shades to capture tone variations from the pixel's red, green and blue values [13]. The average value of the intensity of red, green and blue colors are calculated and taken as the gray level for that pixel. Hence, a gray-scaled image is obtained where each of the pixels in the image contains a set of 2^8 permissible values, which ranges from 0 to 255 [13]. This process is required before the next step, which is MSER because MSER runs on gray-scaled images.

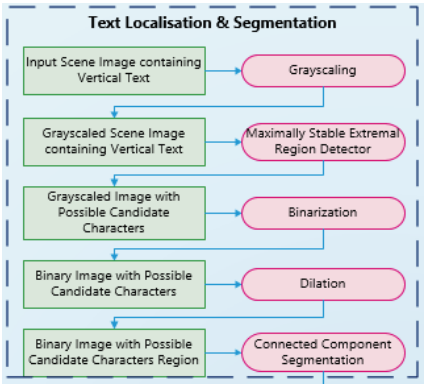


Fig. 5. Text localization and segmentation.

To overcome the illumination for natural scene images, MSER is selected for text localization in the proposed model. MSER is a method to extract stable connected components in a gray-scaled image [1]. MSER captures the stable regions that remain predominantly unchanged for a wide range of threshold values [14]. For a gray-scaled image, thresholding is performed for a range of values starting from 0 to 255. Pixel intensity values below the threshold are set to 'black', and intensity values above, to 'white'. Considering a set of frames consisting of thresholded images corresponding to each threshold increment, there will be a white image in the beginning. Starting from a small seed area, with one or few pixels, the region continues to grow as it fills the object containing the initial seed area corresponding to the local minima intensity. As it grows, some regions corresponding to two local minima are merged together. Fig. 6 shows the process of transformation from a white image to a black image, which is the final image. Extremal region is a connected region for some thresholds that implies the minimum or maximum intensity regions. MSER is an extremal region that is the most stable on the threshold image, which remains unchanged for a wide range of threshold values [15]. Thus, this gives an output of gray-scale image with possible candidate characters. The result obtained is as shown in Fig. 7.

Fig. 6. Sequence of binary images at different value of thresholdings [1].

TABLE I. THE MSER ALGORITHM [16].

Algorithm 1 Maximally Stable Extremal Regions Detector	
Input: grayscale image	
Output: list of maximally stable extremal regions	
1	Pixels in the image are sorted out according to their intensity.
2	For the sorted pixels, place the pixel in the image in an increasing or decreasing order.
3	Update the structure of the connected components.
4	Update the area of the affected connected components.
5	For the connected components, detect the regions that are of local minima based on the rate of change of area function with threshold.
6	Define the detected regions as MSER.

Fig. 7. Input image (left) and result of MSER on gray-scaled image [1].

Fig. 8. Result of MSERs on binary image (left) and after dilation [1].

From the possible candidate characters obtained using MSER, binarization is used to convert the obtained results into a binary image. Binarization converts gray-scaled image to binary image which contains only black and white [17]. The detected regions obtained from MSER is converted into white pixels, while the unwanted regions into black pixels. The process of binarization is necessary to remove the unwanted background. After that, dilation which is also known as "thickening" is used to combine the detected letters in each words in the image into a single region as shown in Fig. 8.

Inspired by Ahmed [1], MSER detector is able to extract stable connected components in the gray-scale image. However, Ahmed [1] used vertical projection profile analysis for text segmentation which has limitation in determining the optimal threshold. Hence, the method of connected component segmentation has been adopted from Shivakumara, et al. [18] as connected component segmentation uses skeletonization which is able to segment scene texts from images and videos and eliminates non-text region. The process of connected component segmentation is shown in Fig. 9.

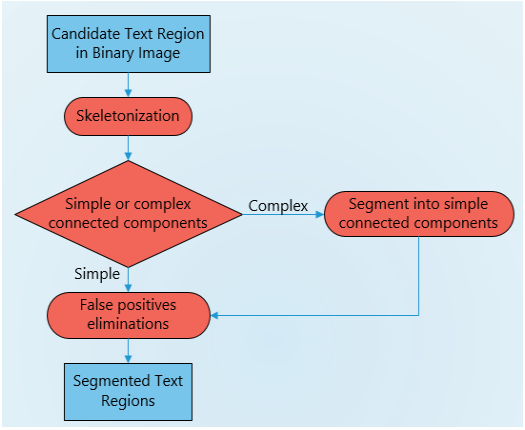


Fig. 9. Flowchart for connected components segmentation [18].

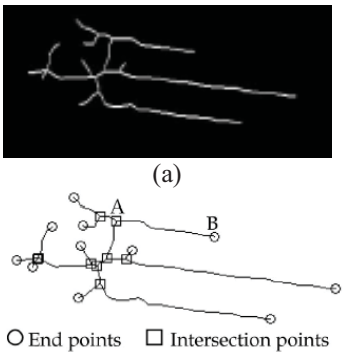


Fig. 10. Image of (a) after skeletonization [18] and (b) complex connected component [18].

Skeletonization reduces the area of foreground regions to a skeletal remnant that shows connectivity of the foreground region in a binary image as shown in Fig. 10(a) [18]. The skeletal remnant shows whether connected components are simple or complex. The connected component is indicated as a simple connected component when there is no intersection but only two end points, else it is indicated as complex connected component. Fig. 10(b) shows an example of complex connected component. In order to segment the complex connected component into multiples of simple ones, the intersection points are eliminated. After all the simple connected components are obtained, it is important to filter false positives. To determine whether the connected component is a true text box or false positive, the straightness and edge density of connected component are calculated as shown in Equation 1 and 2. The connected component is recognized as a true text box when it satisfies Equation 3. Hence, skeletonization is important as it eliminates false positives to increase the accuracy of this proposed modelling.

$$Straightness = \frac{Length\ of\ skeleton}{End\ distance} \quad (1)$$

$$Edge\ density = \frac{Edge\ length}{Area\ of\ connected\ component} \quad (2)$$

$$Straightness < 1.2^{\,Edge\ density} \quad (3)$$

B. Text Recognition

Text recognition takes place after text localization and segmentation. Starting with segmented text regions obtained in binary image, the image undergoes orientation determination and correction and Optical Character Recognition (OCR). Results obtained from OCR is determined before it undergoes string formation for vertical texts. The process of text recognition is described in detail in Fig. 11.

After all the true text boxes have been identified, the model identifies the orientation of the vertical scene texts as horizontal-stacked vertical texts, top-to-bottom vertical texts or bottom-to-top vertical texts. If it is a top-to-bottom vertical text, the text detected will be rotated at 90 degrees, while bottom-to-top vertical texts rotate at -90 degrees as shown in Fig. 12. After text orientation, the text is able to undergo OCR as the texts are already horizontally aligned. However, horizontal-stacked vertical texts do not have to undergo rotation as the letters used in horizontal-stacked vertical texts are already horizontal. The algorithm proposed is shown in TABLE II.

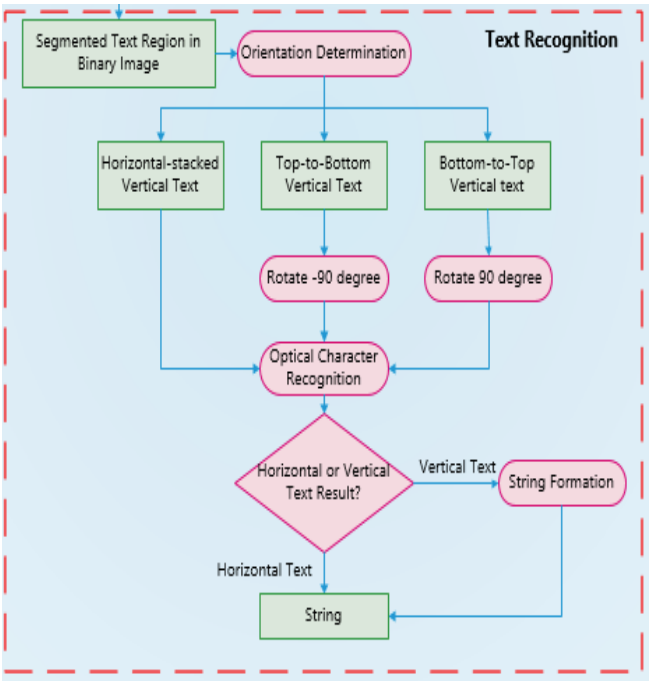


Fig. 11. Process of text recognition.

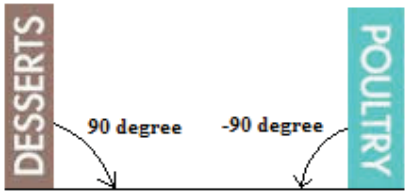


Fig. 12. Rotation for both bottom-to-top and top-to-bottom vertical scene texts.

TABLE II. ALGORITHM FOR ORIENTATION CORRECTION.

Algorithm 2 Orientation correction	
Input: binary image of detected vertical scene text region	
Output: binary image of detected horizontal scene text region	
1	*Load the input image.*
2	*Let Type A = Top-to-bottom vertical text*
3	*Type B = Bottom-to-top vertical text*
4	*Type C = Horizontal-stacked vertical text*
5	***If** input image = Type A*
6	*rotate image at $\theta = 90°$;*
7	***If** input image = Type B*
8	*rotate image at $\theta = -90°$;*
9	***else***
10	*do nothing;*
11	***END***
12	*Load the binary image after rotation.*

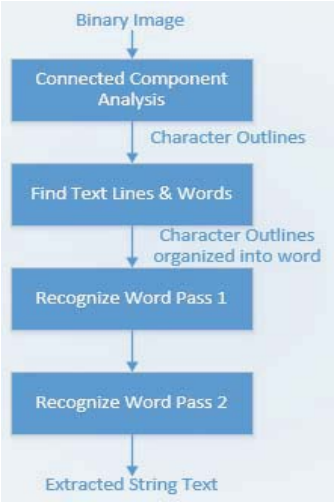


Fig. 13. Flowchart for OCR Process [1].

Optical Character Recognition is a popular method used in text recognition. Today, Tesseract OCR engine has been introduced as one of the most accurate open source OCR available in the internet [1]. The process of OCR is shown in Fig. 13. Based on the texts obtained after text rotation, OCR is able to recognize top-to-bottom and bottom-to-top vertical texts at this stage. As for horizontal-stacked vertical texts, OCR is able to detect the letters letter-by-letter but unable to read them as a word. Therefore, the last process, which is string formation, is required to save the letters detected for vertical-stacked into a new string where the computer is able to recognize it as a word.

IV. Evaluation Metrics for the Proposed Model

Evaluation metrics are adopted to determine the performance of text recognition using three quantitative measures, which are precision, recall and f-measure [19].

Precision evaluates the confidence of the algorithm proposed as it calculates the percentage of the text regions that are correctly detected as compared to the text region claimed [19]. It can also be defined as the total number of true positives divided by the total sum of both true positives and false positives as shown in Equation 4. True positives represent the true predicted text regions, while false positives represent the falsely predicted text regions [19, 20].

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (4)$$

Recall is calculated to determine the sensitivity of the algorithm proposed, by providing a ratio of the number of correct estimates to the total number of targets [21]. Recall can be calculated by dividing the number of true positives by the sum of total of true positives and false negatives as shown in Equation 5. False negatives represent the missed out detection of text region [19, 20].

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (5)$$

As for f-measure, it is the harmonic average, which combines the figure of precision and recall into a single measure of quality. The calculation of f-measure is shown in Equation 6 [21].

$$F\text{-}measure = \frac{2\ x\ Precision\ x\ Recall}{(Precision + Recall)} \quad (6)$$

If an algorithm obtains a high precision value but with low recall value, it shows that the model detects text accurately but not all text regions are detected. In contrary, algorithm with a low precision value but with high recall value, shows the model detects more text regions but also more false positives detection [1]. Therefore, an ideal algorithm would be able to obtain both high precision and high recall values, which indicates a high accuracy of text region detection.

V. Vertical Scene Texts Dataset

A collection of 500 images have been collected for development and evaluation purposes. The images were collected from various websites and also captured from the surroundings. They consist of different types of vertical scene texts, which are horizontal-stacked, top-to-bottom and bottom-to-top vertical texts. Some of the sample images are shown in Fig. 14. The dataset collected is suitable for this research as some of them are obtained from the established datasets such as MSRA-TD500 [7], Street View Test (SVT) [22], NEOCR [23] and more. Besides, the 500 images containing vertical texts will undergo ground truth checking during the performance evaluation of the proposed model.

Fig. 14. Samples of images containing vertical scene texts

VI. Conclusion

Recognition of vertical scene texts could provide additional and useful source of information to understand the natural scenes well. However, most of the existing texts recognizers are lack of the recognition ability for vertical scene texts in natural scenes. Therefore, a model for effective automatic recognition of vertical texts in natural scene images is proposed in this research. Thus, the expected contribution of this research to design and evaluate the proposed scene text recognition model for an effective automatic recognition of horizontal-stacked vertical texts, top-to-bottom vertical texts and bottom-to-top vertical texts.

Acknowledgment

This research work is supported by the Swinburne Melbourne Sarawak Research Collaboration Scheme.

References

[1] A. K. Ahmed, "Signage recognition based wayfinding system for the visually impaired," 2015.

[2] U. Roy, "Text Recognition and Retrieval in Natural Scene Images," International Institute of Information Technology Hyderabad, 2015.

[3] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing,* vol. 27, no. 8, pp. 3676-3690, 2018.

[4] A. Mishra, "Understanding Text in Scene Images," International Institute of Information Technology Hyderabad, 2016.

[5] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE transactions on pattern analysis and machine intelligence,* vol. 37, no. 7, pp. 1480-1500, 2015.

[6] S. Lee and J. H. Kim, "Integrating multiple character proposals for robust scene text extraction," *Image and Vision Computing,* vol. 31, no. 11, pp. 823-840, 2013.

[7] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-Sensitive Regression for Oriented Scene Text Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5909-5918.

[8] A. Sain, A. K. Bhunia, P. P. Roy, and U. Pal, "Multi-oriented text detection and verification in video frames and scene images," *Neurocomputing,* vol. 275, pp. 1531-1549, 2018.

[9] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170,* 2017.

[10] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing,* vol. 23, no. 11, pp. 4737-4749, 2014.

[11] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partition and grouping," *IEEE Transactions on Image Processing,* vol. 20, no. 9, pp. 2594-2605, 2011.

[12] Capture2Text. (2018, 15 August). *CApture2Text*. Available: http://capture2text.sourceforge.net/

[13] G. Kaur and P. Sethi, "A novel methodology for automatic bacterial colony counter," *International Journal of Computer Applications,* vol. 49, no. 15, 2012.

[14] A. V. a. B. Fulkerson. (2007, 23 August). *Extracting MSERs*. Available: http://www.vlfeat.org/overview/mser.html#tut.mser.param

[15] J. Feild, "Improving text recognition in images of natural scenes," 2014.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing,* vol. 22, no. 10, pp. 761-767, 2004.

[17] N. Garg, "Binarization Techniques used for grey scale images," *International Journal of Computer Applications,* vol. 71, no. 1, 2013.

[18] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A laplacian approach to multi-oriented text detection in video," *IEEE transactions on pattern analysis and machine intelligence,* vol. 33, no. 2, pp. 412-419, 2011.

[19] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.

[20] P. Sahare and S. B. Dhok, "Review of text extraction algorithms for scene-text and document images," *IETE Technical Review,* vol. 34, no. 2, pp. 144-164, 2017.

[21] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003, pp. 682-687.

[22] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*, 2010, pp. 591-604: Springer.

[23] R. Nagy, A. Dicker, and K. Meyer-Wegener, "NEOCR: A configurable dataset for natural image text recognition," in *International Workshop on Camera-Based Document Analysis and Recognition*, 2011, pp. 150-163: Springer.