Detecting Text in Manga using Stroke Width Transform

Boonyarith Piriyothinkul*, Kitsuchart Pasupa†
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520
Thailand
Email: *58070077@kmitl.ac.th, †kitsuchart@it.kmitl.ac.th

Masanori Sugimoto
Graduate School of Information Science & Technology
Hokkaido University
Hokkaido 060-0808
Japan
Email: sugi@ist.hokudai.ac.jp

***Abstract*—The Japanese comic-book style known as manga is becoming a popular topic for researchers. This paper focuses on the problem of detecting text regions in manga pages. Because it is time-consuming and laborious to identify the text regions in images manually, an automatic approach is highly desirable. Here, we propose a new text-detection method for manga using a Stroke Width Transform (SWT) technique in conjunction with a Support Vector Machine (SVM). Conventional SWT-based text-detection techniques perform poorly with manga because both text and non-text objects have similar characteristics for strokes, lines, and shapes. To better suit manga, we propose modifying the rules for finding letter candidates, which improves the ability to capture text. An SVM is then used to classify image patches into letter and nonletter regions. We compared our proposed framework with a conventional framework and other text-detection methods including deep-learning techniques. In the results, our proposed method achieved the highest F-measure of 0.506.**

I. Introduction

Japanese comics are well known across the world as "manga." There has been extensive research in manga [1]–[7] and benchmark datasets for this research, including Manga109 [8]. Manga109 contains more than 20,260 manga pages collected from 109 volumes and drawn by professional manga artists in Japan. The annotations in the dataset refer to face, body, frame, and text regions, with the text regions being identified and annotated manually. The annotations were undertaken by human without using any support from automation. This task was time-consuming and laborious. Therefore, an automatic annotation system is highly desirable.

In comparison to regular artworks, manga contains much more text and this research focuses on manga. Several text-detection methods for manga have been proposed [9], [10] but these methods are limited to specific layouts, dialogue-balloon positions, and art frames. Moreover, the proposed frameworks are not fully automated and robust to the various text styles present in manga. Recently, a deep-learning technique, the convolutional neural network, was applied to the feature-extraction aspect of text detection in manga [11]. It performed well and was able to reduce the false-positive rate without limitations on layout structure. However, its use of a deep-learning technique results in a high computational cost [11].

In this research, we focus on developing a framework for text detection that is robust to manga components. The Stroke Width Transform (SWT) technique is used to describe the characteristics of a component "stroke." It was proposed initially as part of a framework for detecting text in natural scenes, based on the assumptions that there would be high-density edges to text areas and smooth backgrounds [12]. However, employing the framework proposed by [12] to detect text in manga can generate many false-positive text predictions that reduce performance. This is because manga involves greyscale image and object details such as size, stroke, and background that are similar to text. We therefore aim to adopt an SWT similar to that proposed by [12] and improve its framework to suit manga.

The proposed technique can be divided into four main steps: (i) the Stroke Width Transform, (ii) finding letter candidates, (iii) letter classification, which aims to reduce false-positive results using the Support Vector Machine (SVM) with the histogram of oriented gradients (HOG) as a feature, and (iv) grouping letters into lines.

The paper is structured as follows. Detecting text in natural scenes with the SWT [12] is briefly explained in Section II. Our proposed framework is described in Section III. Section IV discusses our experiments and their results. Our conclusions are contained in Section V.

II. Detecting Text in Natural Scenes with the SWT

A. The Stroke Width Transform

The SWT is a text-detection technique for photographs that uses the "stroke" features of an object [12], which can differentiate between text and nontext objects. The extraction process begins with the output image being initialized (by assigning ∞ to all its pixels). The output image is the same size as an input image. Next, the Canny edge detection [13] is employed to extract the edges objects in the image. The distance between the borders of a stroke can be calculated by considering each edge pixel p and finding a corresponding pixel q on the other edge. As shown in Fig. 1(b), the gradient direction of edge pixel p (d_p) points to edge pixel q. If d_p is approximately opposite d_q, $d_q = -d_p \pm \pi/6$ and any pixels that are located in between p and q will be assigned to $\|\overrightarrow{p-q}\|$, as shown in Fig. 1(c). If the pixel already contains a width $\|\overrightarrow{p-q}\|$ value and the new value of $\|\overrightarrow{p-q}\|$ is smaller than the existing one, it will be replaced by the smaller value.

Otherwise, it remains the same. Eventually, a matrix with stroke-width values assigned to each element will be returned.

B. Finding letter candidates

In this step, the SWT output is used to find letter candidates by eliminating unrelated components. Each pixel is checked and compared with its neighbors. If their stroke-width ratio does not exceed 3.0, they will be grouped together. However, if the connected components are too big or too small, they will be eliminated according to these two rules: (i) the ratio of the diameter of the connected component to its median stroke width must be less than 10 and (ii) its height must be between 10 and 300 pixels, as shown in (1).

$$f(d,h,\tilde{s}) = \begin{cases} 1, & \text{if } \frac{d}{\tilde{s}} < 10 \text{ and } 10 < h < 300 \\ 0, & otherwise \end{cases}, \quad (1)$$

where d is the diameter of the connected component, h is its height in pixels, and $\tilde{s}$ is its median stroke width.

C. Grouping letters into text lines

The letter candidates will be grouped into lines based on the distance between letters, the stroke-width value, and their height. Two letter candidates can be grouped together if (i) their median stroke-width ratio is less than 2.0, (ii) their height ratio does not exceed 2.0, and (iii) the distance between them does not exceed three times that of the wider candidate. After grouping, all letter chains can be merged to form a word. Chains can be merged if two chains have shared letters and the same direction. This step terminates when there are no more chains to be merged. Finally, the group or chain for a word is obtained from the image.

III. Proposed Method

In this section, we describe our proposed method, which utilizes an SWT in conjunction with an SVM and a HOG. As mentioned in Section I, the purpose of the SWT is to detect text in natural scenes. Unfortunately, when an SWT is simply applied to manga, it can lead to many false-positive results, as shown in Fig. 2. This is because of the characteristic differences between objects in manga and in the real world. Moreover, various graphic components in manga are very similar to text characters, such as grass, a character's hair, and background details, as shown in Figs. 2(a) and 2(b). Our proposal therefore modifies the steps involving finding letter candidates and grouping letters and adds a new step to make the approach more suited to manga, as shown in Fig. 3. In the last step, it utilizes an SVM to reduce the false positives in the previous finding-letter-candidates step.

A. The Stroke Width Transform

This step is similar to that proposed in [12]. The input is a manga image that has been transformed by the SWT operator. The size of the output image is the same as the input.

B. Finding letter candidates

In manga, there may be various text sizes to compare with the natural scenes. Therefore, we modify the conditions for finding letter candidates to enhance the performance, as shown in (2).

$$f(d,h,w,\tilde{s}) = \begin{cases} 1, & \text{if } 1 < \frac{d}{\tilde{s}} < 15 \text{ and } \tilde{s} \leq 80 \text{ and} \\ & 5 < h, w < 50 \\ 0, & otherwise, \end{cases} \quad (2)$$

where w is the width of the connected component in pixels.

Finally, we obtain the letter candidates shown in Fig. 4. Note that the baseline method fails to capture all the letters while our proposed method captures them all. However, there are many nonletter image patches. Their elimination is the purpose of the subsequent step.

C. Letter classification

In this step, we aim to classify image patches obtained from the previous step into letter and nonletter patches. This will reduce the number of false positives in the letter candidates. We adopt an SVM to perform this task. An SVM is a supervised learning technique that is well known and widely used for classification and regression tasks [14]. The dataset comprises two classes of letter (positive) and nonletter (negative) image patches created from Manga109 [8], as shown in Fig. 5. More details will be described in Section IV. Image-patch features are extracted by using the HOG [15], which is a descriptor for the gradient-direction patterns of objects in terms of their distribution. In this work, a HOG feature is represented by a 2,916-dimensional vector. After this process, the letter patches are grouped into lines in the next step.

D. Grouping letters into lines

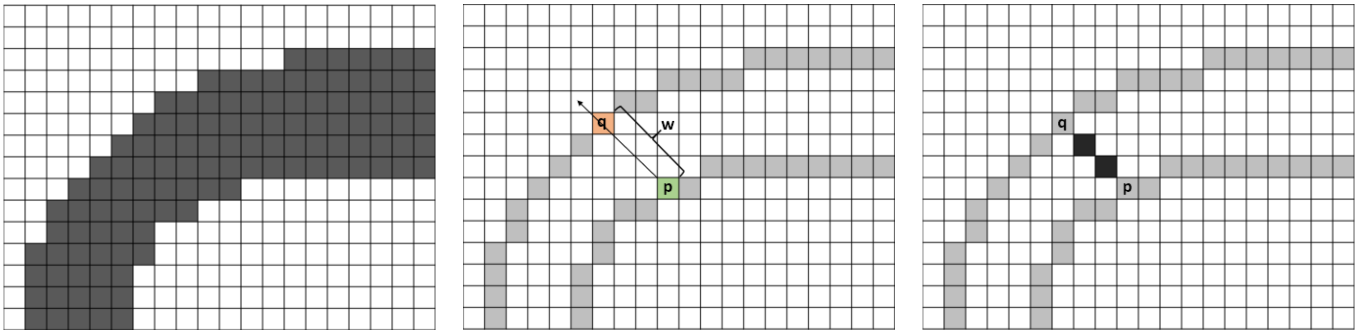
The conventional (baseline) approach [12] groups the letter candidates into lines by using rules about height and median stroke-width values, as explained in Section II. It aims to remove scattered noise, i.e., the nonletter image patches. However, our approach has already dealt with nonletter image patches by using an SVM in conjunction with a HOG. Therefore, we use only the distance between letters to form groups.

Our grouping method uses the classified letters to create lines of letters. A line is created by including in the group any character that is closer to another than 1.5 times the narrower character's width. Characters spaced out less than this are considered as part of the same line, as shown in Fig. 6. Moreover, each line must contain at least three characters and have an area (width $\times$ height) exceeding 2,550 pixels, according to the experiments using our dataset. Finally, the letters are grouped into lines.

IV. Experiments

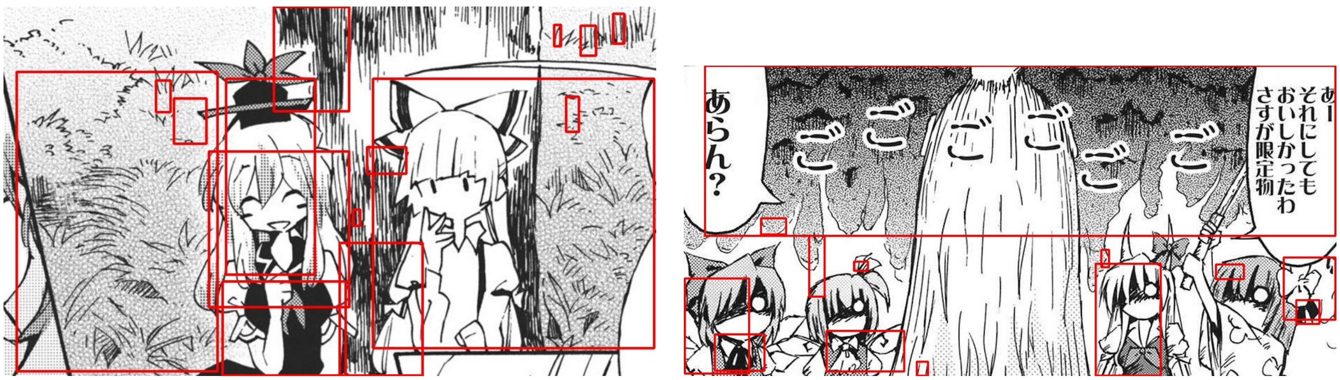
A. Dataset

We used images from Manga109 [8], which comprises 109 annotated volumes of manga created by the Aizawa Yamasaki Laboratory at the University of Tokyo. All manga volumes in the dataset were drawn by professional manga artists in



(a) (b) (c)

Fig. 1: (a) Grey pixels representing a stroke, (b) Pixel p is considered to be at the edge of the stroke, with the gradient direction d_p of p pointing to q, the corresponding pixel on the other edge. w is the distance between the two edge pixels. (c) The pixels along the line connecting p and q are assigned the distance w between p and q.



(a) (b)

Fig. 2: Example of results generated by a conventional SWT method (the baseline method), where there are many false positives because the characteristics of text and graphic elements are very similar [12]. (a) Artist: Shinoasa (b) Artist: Kousei (Public Planet)

Japan and were available commercially to the public between the 1970s and 2010s. Each page is annotated with text-region areas suitable for training and evaluating our proposed method.

B. Experimental settings

We used experimental settings similar to those of Aramaki et al. [11], enabling us to make direct comparisons between their results and our results. We selected 100 pages of training data and 100 pages of test data at random from six manga titles, i.e., *Aosugiru Haru, Arisa 2, Bakuretsu Kung Fu Girl, Dollgun, Love Hina, and Uchiha Akatsuki EvaLady*.

Because our approach requires us to build a model for classifying image patches as letters or nonletters, as shown in Fig. 5, we created a dataset comprising 5,201 letter patches and 5,201 nonletter patches. These image patches were obtained from 100 pages selected at random from the Manga109 data set. The data were partitioned into 50 % for training and 50 % for validation. We used an SVM with a radial basis function kernel. This has two hyperparameters that require tuning, i.e., the regularization (C) and kernel (γ) parameters. A grid search was performed, with C and γ ranging from 2^{-10} to 2^{10}, to train the model and evaluate its F-measure on the validation set. We performed the search using the Google Compute Engine n1-highcpu-8. The optimal C and γ values, returning the best F-measure, were 2^5 and $2^{-6.75}$, respectively. This optimized model was used in the letter-classification step for our proposed method.

Our evaluation method for text detection was the same as for the ICDAR 2013 Robust Reading Competition [16]. If the ratio of the overlapped area to the ground-truth area is greater than t_p and the ratio of the overlapped area to the detected region is greater than t_r, correct detection has been achieved. Here, t_p and t_r were set to 0.5, as for Aramaki et al. [11]. Precision and recall can be calculated by (3) and (4), respectively.

$$P = \frac{\text{\#Correctly Detected Rectangles}}{\text{\#Detected Rectangles}} \quad (3)$$

$$R = \frac{\text{\#Correctly Detected Rectangles}}{\text{\#Rectangles of the Ground-truth}} \quad (4)$$

The F-measure can then be calculated by

$$F = 2 \cdot \frac{P \cdot R}{P + R}. \quad (5)$$

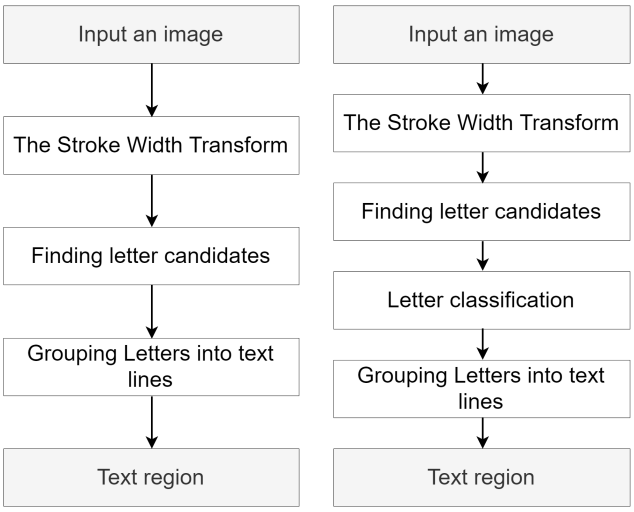


Fig. 3: The flowchart for the baseline method [12] comprises three steps whereas our proposed method has four steps: (a) baseline method and (b) proposed method.

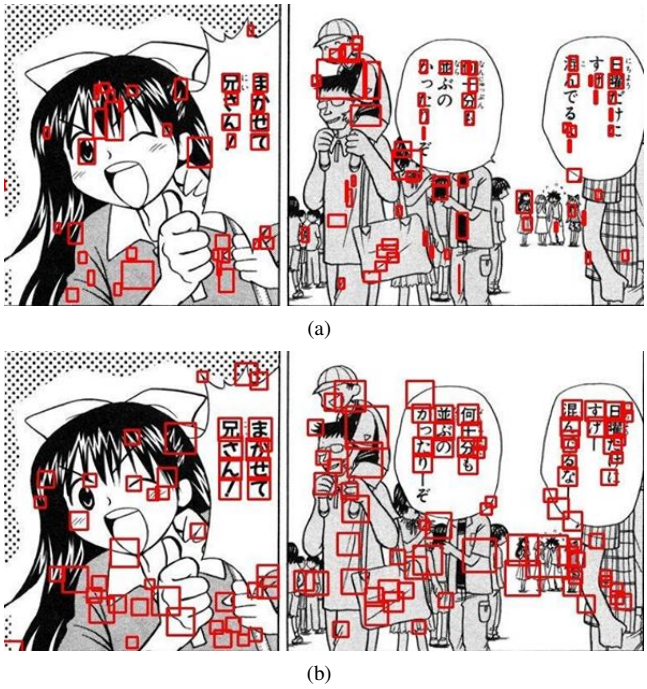


Fig. 4: The results of generating letter-candidate regions in Arisa ©Yagami Ken. The image patches with bounding boxes are the letter-candidate regions: (a) baseline method and (b) proposed method.

We compared our proposed method with a conventional (baseline) SWT method [14] and with previous published results for this dataset using various text-detection methods reported in literature. These other approaches were the Basic Grouping+ImageNet Classification model (BG+ImN) [11], the Basic Grouping+Illustration2Vec model (BG+I2V) [11],

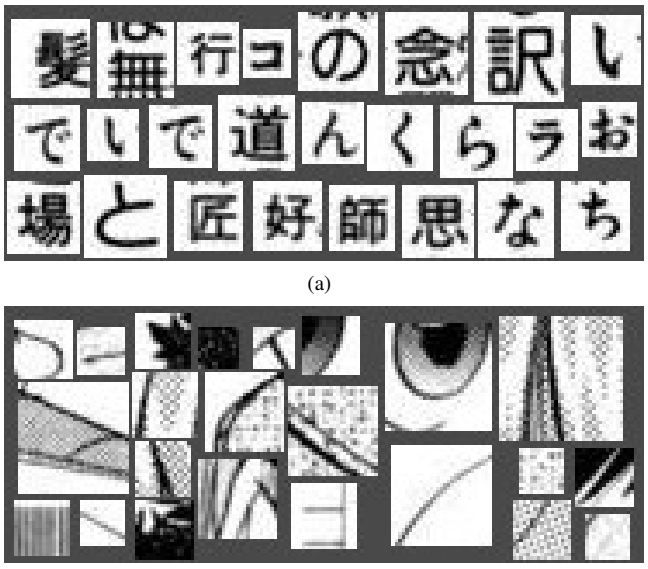


Fig. 5: Examples of letter image patches: (a) positive image patches and (b) negative image patches.

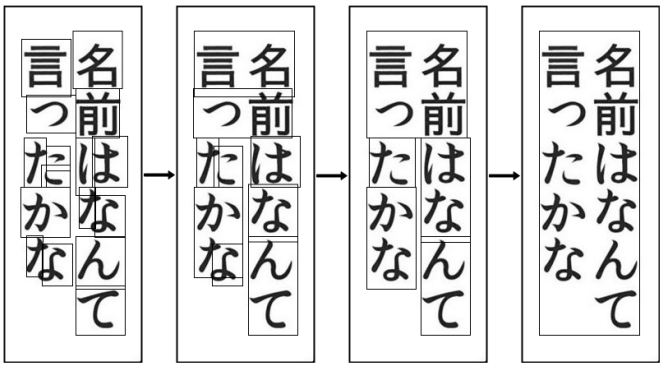
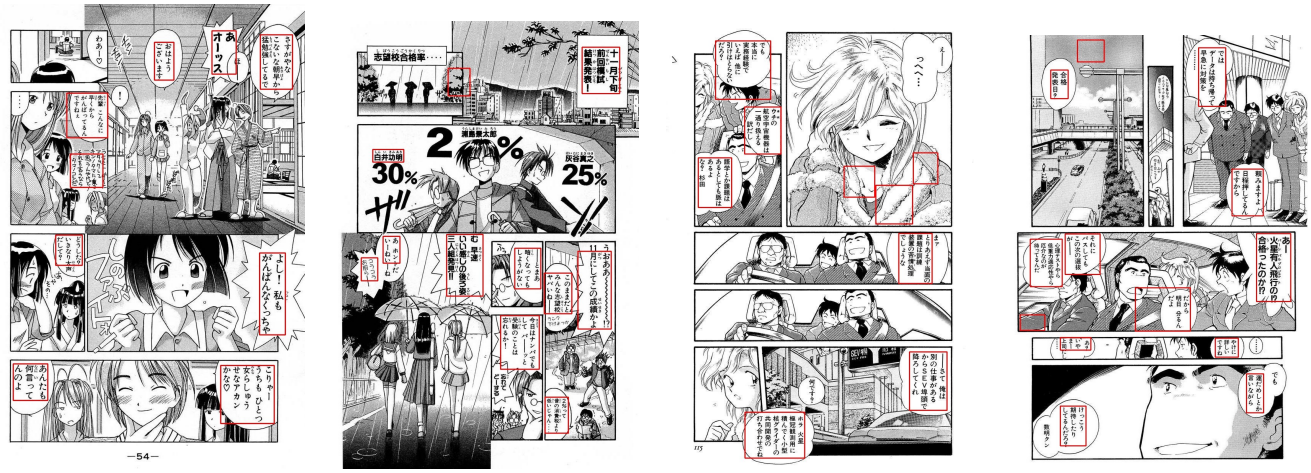


Fig. 6: An example of the process of grouping letters into lines.

Scene Text Detection (STD) [17], Speech Balloon Detection (SBD) [9], and Text Line Detection (TLD) [18]. These approaches use a variety of text-detection methods, including heuristic assumptions (direction of text, layout, dialogue balloon) and convolutional neural networks. The results are shown in Table I. Note that our proposed method yields the highest F-measure of 0.506. The proposed method clearly outperforms the baseline method for all performance measures. Moreover, its F-measure is higher than that for both BG+ImN and BG+I2V. Note also that both BG+ImN and BG+I2V utilize deep-learning methods. However, the best precision and recall is achieved by BG+I2V and BG+ImN at 0.715 and 0.481, respectively. Examples of text regions detected by our proposed technique are shown in Fig. 7.

It is interesting that our method can perform better than deep-learning techniques. This might be because BG+ImN uses the ImageNet Classification model [19], which was trained using a large number of real-world object images. However, manga objects are different from objects in natural



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

Fig. 7: Examples of text regions detected by our proposed method on (a–b) Love Hina © Ken Akamatsu and (c–d) Eva Lady © Miyone Shi.

TABLE I: The experimental results with Manga109 [8]. The proposed method is compared with the baseline method [12], the Basic Grouping+ImageNet Classification model (BG+ImN) [11], the Basic Grouping+Illustration2vec model (BG+I2V) [11], Scene Text Detection (STD) [17], Speech Balloon Detection (SBD) [9], and Text Line Detection (TLD) [18].

Method	Precision	Recall	F-measure
STD [17]	0.165	0.051	0.078
SBD [9]	0.180	0.102	0.130
TLD [18]	0.095	0.095	0.095
BG + ImN [11]	0.451	**0.481**	0.466
BG + I2V [11]	**0.715**	0.191	0.301
Baseline [12]	0.068	0.336	0.113
Our method	0.564	0.458	**0.506**

scenes. Another deep-learning model is BG+I2V. Although it achieved the best precision in our experiments, its recall is lower than both BG+ImN and our proposed technique. This method uses Illustration2Vec [20] as a classification model. It was trained using a dataset of manga and anime illustrations (the Japanese animation art) from several sources (e.g., Danbooru and Safebooru) that is similar to our approach. The model was not built specifically for text classification but for other purposes such as tag prediction and finding similar images. This might explain why the model appeared relatively unsuccessful in our experiments.

V. Conclusion

In this research, we have proposed a text-detection framework for manga using an SWT in conjunction with an SVM and a HOG. Experiments were conducted using a well-known Manga109 dataset. The proposed method yielded the best F-measure in comparison with a baseline method and those involving deep learning. To date, we have only tested our method on Japanese comics, where it works well. However, more investigations are required to find additional improvements and we aim to extend the method to other languages.

Acknowledgment

We would also like to show our gratitude to Xiaoyong Zhang, Sendai National College of Technology for sharing their wisdom. We also thanks to Napassorn Thammaviwatnukoon and ZUN for inspiration on this research. This research is supported by King Mongkuts Institute of Technology Ladkrabang.

References

[1] H. Yanagisawa, T. Yamashita, and H. Watanabe, "A study on object detection method from manga images using CNN," in *Proceedings of the International Workshop on Advanced Image Technology (IWAIT 2018)*, Chiang Mai, Thailand., Jan 2018, pp. 1–4.

[2] X. Liu, C. Li, H. Zhu, T.-T. Wong, and X. Xu, "Text-aware balloon extraction from manga," *The Visual Computer*, vol. 32, no. 4, pp. 501–511, Apr 2016. [Online]. Available: https://doi.org/10.1007/s00371-015-1084-0

[3] X. Pang, Y. Cao, R. W. Lau, and A. B. Chan, "A robust panel extraction method for manga," in *Proceedings of the 22nd ACM International Conference on Multimedia (MM 2014)*. New York, NY, USA: ACM, 2014, pp. 1125–1128. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654990

[4] Y. Aramaki, Y. Matsui, T. Yamasaki, and K. Aizawa, "Interactive segmentation for manga using lossless thinning and coarse labeling," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA 2015)*, Hung Hom, Kowloon, Hong Kong, Dec 2015, pp. 293–296.

[5] T. Ogawa, A. Otsubo, R. Narita, Y. Matsui, T. Yamasaki, and K. Aizawa, "Object detection for comics using manga109 annotations," *CoRR*, vol. abs/1803.08670, 2018. [Online]. Available: http://arxiv.org/abs/1803.08670

[6] S. Kovanen and K. Aizawa, "A layered method for determining manga text bubble reading order," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2015)*, Quebec City, QC, Canada, Sept.

[7] Y. Matsui, T. Shiratori, and K. Aizawa, "Drawfromdrawings: 2D drawing assistance via stroke interpolation with a sketch database," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1852–1862, 2017.

[8] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications*, vol. 76, no. 20, pp.

21 811–21 838, Oct 2017. [Online]. Available: https://doi.org/10.1007/s11042-016-4020-z

[9] H. Tolle and K. Arai, "Manga content extraction method for automatic mobile comic content creation," in *Proceedings of the International Conference on Advanced Computer Science and Information Systems (ICACSIS 2013)*, Bali, Indonesia, Sept 2013, pp. 321–328.

[10] C. Rigaud, T. Le, J. . Burie, J. Ogier, S. Ishimaru, M. Iwata, and K. Kise, "Semi-automatic text and graphics extraction of manga using eye tracking information," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, Santorini, Greece, April 2016, pp. 120–125.

[11] Y. Aramaki, Y. Matsui, T. Yamasaki, and K. Aizawa, "Text detection in manga by combining connected-component-based and region-based classifications," in *Proceedings of the IEEE International Conference on Image Processing (ICIP 2016)*, Phoenix, AZ, USA, Sept 2016.

[12] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA, USA, Jun 2010, pp. 2963–2970.

[13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[14] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, Jun 1999. [Online]. Available: https://doi.org/10.1023/A:1018628609742

[15] W. T. Freeman, W. T. Freeman, M. Roth, and M. Roth, "Orientation Histograms for Hand Gesture Recognition," in *Proceedings of the International Workshop on Automatic Face and Gesture Recognition*, 1994, pp. 296–301. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.618

[16] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i. Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazn, and L. P. de las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*, Kolkata, India, Aug 2013, pp. 1484–1493.

[17] L. Gómez and D. Karatzas, "Multi-script text extraction from natural scenes," in *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, Washington, DC, USA, Aug 2013, pp. 467–471.

[18] C. Rigaud, D. Karatzas, J. Van De Weijer, J.-C. Burie, and J.-M. Ogier, "Automatic text localisation in scanned comic books," in *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, Barcelona, Spain, Feb 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00841492

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012)*, vol. 1. Lake Tahoe, Nevada, USA: Curran Associates Inc., Dec 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[20] M. Saito and Y. Matsui, "Illustration2Vec: A semantic vector representation of illustrations," in *SIGGRAPH Asia 2015 Technical Briefs*. Kobe, Japan: ACM, Nov 2015, pp. 5:1–5:4. [Online]. Available: http://doi.acm.org/10.1145/2820903.2820907