SickZil-Machine: A Deep Learning Based Script Text Isolation System for Comics Translation

U-Ram Ko and Hwan-Gue Cho(✉)

Department of Electrical and Computer Engineering, Pusan National University, Pusan, South Korea
{rhdnfka94,hgcho}@pusan.ac.kr

Abstract. The translation of comics (and Manga) involves removing text from a foreign comic images and typesetting translated letters into it. The text in comics contain a variety of deformed letters drawn in arbitrary positions, in complex images or patterns. These letters have to be removed by experts, as computationally erasing these letters is very challenging. Although several classical image processing algorithms and tools have been developed, a completely automated method that could erase the text is still lacking. Therefore, we propose an image processing framework called 'SickZil-Machine' (SZMC) that automates the removal of text from comics. SZMC works through a two-step process. In the first step, the text areas are segmented at the pixel level. In the second step, the letters in the segmented areas are erased and inpainted naturally to match their surroundings. SZMC exhibited a notable performance, employing deep learning based image segmentation and image inpainting models. To train these models, we constructed 285 pairs of original comic pages, a text area-mask dataset, and a dataset of 31,497 comic pages. We identified the characteristics of the dataset that could improve SZMC performance. SZMC is available at: https://github.com/KUR-creative/SickZil-Machine.

Keywords: Comics translation · Deep learning · Image manipulation system

1 Introduction

Comic literature (or Manga) is globally appreciated. Economically, its market has been growing, especially for the digital form comics [3]. However, the automatic translation of comics is difficult, due to their inherent characteristics. Currently, most comics are translated manually.

Translation of comics could be divided into two phases. First, erasing the foreign language text (Fig. 1-a), and filling the erased areas with a picture to match with their surrounding regions. Second, The translated text is either typeset (font letters) or drawn (calligraphic letters) on the image (Fig. 1-b). Because

X. Bai et al. (Eds.): DAS 2020, LNCS 12116, pp. 413–425, 2020.
https://doi.org/10.1007/978-3-030-57058-3_29

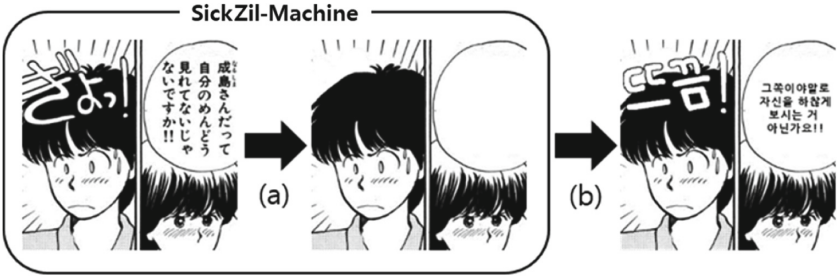

Fig. 1. Example of Japanese to Korean translation of comics. (a) Text removal from the foreign comics (b) Drawing the translated text. SickZil-Machine (SZMC) automates text removal for the comics translation. Original image was extracted from Manga109 dataset [8], ©Yoshi Masako.

the text in comics is drawn at arbitrary places, and often on complex images or patterns, erasing the text from the original comic image is time-consuming, and labor-intensive. The text found in comics include printed font letters and artist-specific handwritten calligraphies, which could be of various styles and sizes. Therefore, detection and segmentation of such text is challenging. Consequently, removing such text, using the classical image processing algorithms, is difficult. To solve this, we propose a deep learning based framework "SickZil-Machine (SZMC)" for effective translation. "SickZil" in Korean means comics editing task.

2 Proposed Approach

Adobe Photoshop (Adobe Inc., San Jose, USA) has been used by professional editors to remove the text in comics. The editors commonly use several classical image processing algorithms that are provided in Photoshop. Although the macros system in Photoshop is useful in removing simple text, it requires the editors' manual intervention for the text in more complex backgrounds.

We approach the removal of comics text as a problem that is combined image segmentation and image inpainting (Fig. 2). SZMC segments the text areas in the comics image (Fig. 2-a) using the image segmentation model, erases the segmented text area, and fills the erased area (Fig. 2-b) using the image inpainting model.

An alternative approach was proposed, which detected and erased the text in an image using a single end-to-end neural network model, known as EnsNet [17]. Although EnsNet had faster processing speed and required lower memory, it had several limitations. Obtaining sufficient data was not feasible, as having comic image pairs, one with the text and the other without, was rare. Further, the majority of them are proprietary data and not public. Second, as the removable text area was implicitly determined by the end-to-end model, determining the erasable area by the user was not possible.

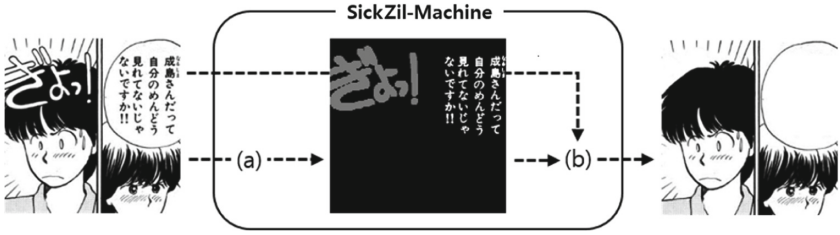

Fig. 2. Proposed approach for automatic text removal during comics translation. (a) image segmentation (b) image inpainting. SZMC segmented the text areas and erased them. The erased area was drawn using image inpainting model, to match with the surrounding. Original image was extracted from Manga109 dataset [8], ©Yoshi Masako.

3 Related Work

3.1 Previous Researches on Comics Image Analysis

Although segmentation of natural or medical images has been extensively studied, the automatic segmentation of text areas from comics has not yet been investigated. Segmentation of common objects, such as lines, speech bubbles, and screen tones was studied [5,7]. Further, detection and analysis of text, in bounding boxes, has been reported [2,11]. [16] was proposed as an open source project for segmenting the text area at the pixel level (Table 1).

Table 1. Summary of the related work on comic content analysis

No	Input	Output	Method	Ref.
1	Manga page	Region	Graph-cut with user stroke	[7]
2	Manga frame	Lines	LoG and Gaussian filter	[5]
3	Manga page	Text region	Connected-Component analysis	[2]
4	Comics text	Plain text	Segmentation-free learning	[11]
5	Comics page	Text mask	Learning-based (MobileNet)	[16]
6	Comics page	Text mask	Learning-based (U-net)	[14]

3.2 Previous Researches on Generic Text Eraser

To the best of our knowledge, we cannot find end-to-end method for removing text from comics. STEraser [10], EnsNet [17], and MTRNet [13] try to remove text from natural images in order to erase personal private information such as telephone numbers, home addresses, etc. Unlike the proposed two-step method, STEraser and EnsNet consider text removal as an image transformation task.

STEraser, a first scene text eraser using a neural net, applies U-net-like [12] structure with residual connection instead of concatenation. It trains and inferences with sliding-window-based 64×64 sized crops from the input image. Since STEraser cannot grasp context of the whole image, it cannot erase large text properly.

Unlike STEraser, EnsNet uses the entire image as an input. It follows cGAN [9] framework with novel lateral connection in the generator, and applies multiple losses. Though EnsNet produces a plausible output compare to STEraser, but it is not suitable for the comics translation process. In comics translation, the erasing text depends on the comics editor's decision. Also, those image transformation approach requires input image and the corresponding image with text removed. One serious problem is that such data is hard to collect and very difficult to create.

MTRNet is an inpainting model using modified cGAN. Applying two-step method to remove text, it receives an input image concatenated with a mask that annotates the text location when training and inference. Therefore, a comics editor can select the text to be erased. However, MTRNet is still not applicable to comics translation because it does not consider to erase very large text. In comics, not only small regular letters, but also very large-sized calligraphic characters can exist on a complex background.

We propose two-step approach in order to allow the editor to select the text to erase. To erase large text in comics, we apply more general inpainting model [15] that is not limited to text.

3.3 Previous Researches on Image Inpainting

Adobe Photoshop's content aware fill function has been used by the editors, to erase smaller-sized text in comics. Although this function was effective in erasing small text that was drawn over simple patterns, it failed to naturally erase the text on either the non-stationary images or complex patterns. Further, this function failed to erase large handwritten calligraphic letters, as the classic inpainting techniques disregard the image semantics.

[4] was the first deep learning based end-to-end model for image inpainting. But this model required a longer training period, could be trained only with the square masks, and therefore, overfitted to the square masks. [6] was proposed to reduce the training period and was able to prevent the square mask overfitting. This model was able to acquire the masks as channels in the input image and could inpaint the image by updating the mask for each partial convolution layer, in the feed-forward path. [15] improved [6] by substituting the gated convolution layers for the partial convolution layers. While the partial convolution layers updated the mask channels in the binary values of 0 and 1 (hard gating), the gated convolution layers added the training parameters, for soft gating the mask, to the real values of 0.0 to 1.0.

3.4 Available Comics Datasets

Danbooru2018 [1], and Manga109 [8] are the publicly available comic literature (and Manga) datasets. While Danbooru2018 dataset tags illustrations and comic pages with image-feature metadata, the Manga109 dataset tags the pages from 109 titles of Manga, with the metadata of the script text, the area of the character's face, and the position and contents of the text. However, the mask data indicating a text area at the pixel level, was lacking.

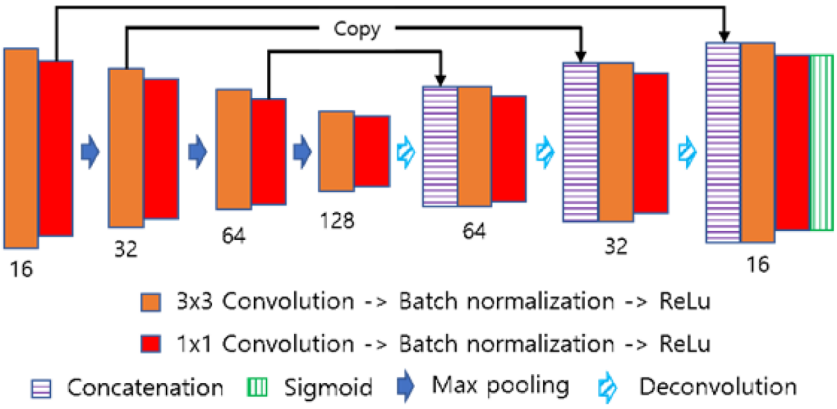

Fig. 3. Modified U-net model for text area segmentation.

4 The Proposed Model

The SZMC segments the text areas to be erased, using a modified U-net [12] and removes the text naturally using [15].

4.1 Step 1 - Text Area Segmentation

U-net is composed of layers that repeat certain basic unit blocks [12]. We set up the U-net basic block with 3×3 and 1×1 filtered convolution layers, followed by a batch normalization layer and a ReLU layer. In our four-layer U-net model, the first, second, third and fourth convolution layers had 16, 32, 64, and 128 filters. We employed maxpooling and deconvolution at the encoder and decoder, respectively (Fig. 3). As the comics text component masks are very unbalanced in classes (background pixels are much more than texts), we used weighted binary cross entropy and Jaccard loss, to train the model.

4.2 Step 2 - Comics Image Inpainting

Comics have not only font-based text, but also calligraphic text drawn by the artist. They can be very difficult to remove naturally due to its large area and irregular shape. Therefore [15], the state-of-the-art image inpainting model, was applied. This model required variously shaped masks in training for best performance. So we used the text components, which were extracted from our comic literature datasets, as mask data.

5 Data Preparation

5.1 Raw Comic Literature Data Collection

Danbooru is a website with anonymously uploaded comics, manga and illustrations. Danbooru2018 dataset is organized by crawling Danbooru [1]. We created a database of selected monochrome-tagged images from Danbooru2018, resulting in a collection of high-quality comics and manga images. Moreover, typical black and white manga images were obtained from Manga109 [8].

5.2 Organizing Dataset for the Image Inpainting Model

The text that could be easily erased with the existing image editing tools, such as that in the speech bubble in Fig. 2, was designated as easyT. However, the text on complex backgrounds that are difficult to erase, such as the calligraphic letters drawn over the man's head in Fig. 2, were designated as hardT. The images in the original comic literature databases were classified into four categories, according to the location of easyT and hardT text: data-1: Images without text, data-2: Images with only easyT text, data-3: Images with easyT and hardT text, data-4: Images with only hardT text.

Two datasets were constructed to train the image inpainting model. The "NoText" dataset was constructed from 20,033 images from data-1, among which 16,033 images were used for training, and 4,000 images were used for testing. The "HasText" dataset was constructed with 27,497 images from datasets-3,4. The NoText test dataset was used for testing the entire system. Since the applied image inpainting model [15] was trained in an unsupervised manner, a validation dataset was not required.

5.3 Dataset Creation for Text Area Segmentation

The dataset for image segmentation consisted of pairs of original comic images and answer masks that could segment the text of the original comic images. "Split" dataset was created to separate the text into two classes: calligraphic and font texts. However, the "All" dataset grouped all the text in one class. The number of images used for training, validation, and testing from both datasets was 200, 57, and 28, respectively.

We used a part of the dataset for image inpainting, to generate the answer image-mask pair data. After clearing easyT from data-2, we generated the answer masks by computing the difference between the original and text-removed images. However, this method could not be applied to create a segmentation mask that had hardT. Thus, the hardT areas in data-3,4 were segmented manually to generate the HardT dataset (Table 2). The first 50 masks were created manually, using GNU Image Manipulation Program. The remaining 235 masks were created by modifying the trained segmentation model output. To prevent the contamination of the validation data, the experimental models were trained separately.

Table 2. Prepared datasets for the system

Model	Name	Train/valid/test	Remarks
Segmentation	Split	200/57/28	Calligraphy, font text separated
Segmentation	All	200/57/28	No separation of text
Inpainting	NoText	16,033/NA/4000	Danbooru2018, Manga109
Inpainting	HasText	27,497/NA/4000	Danbooru2018, Manga109

6 Experiments Results

6.1 Evaluation of Image Segmentation Model

We applied mean intersection over union (IoU) to quantitatively evaluate the image segmentation model. IoU similarity was defined as

$$IoU(G, S) = \frac{G \cap S}{G \cup S},$$

wherein G and S denoted the answer and result sets, respectively.

Table 3. Evaluation of the image segmentation model

Model	Loss	mIoU	Time (sec)
TSII[16]	wbce	0.279	**0.442**
U-net (Split)	wbce	0.424	0.510
U-net (All)	wbce	0.570	0.511
U-net (Split)	Jaccard	0.479	0.511
U-net (All)	Jaccard	**0.602**	0.512

The SZMC was evaluated using an Intel i7 CPU with gtx1070ti GPU. SZMC exhibited slightly slower performance compared to TSII [16]. Notably, SZMC

exhibited more than double IoU similarity. Moreover, IoU similarity, for text, of the All dataset trained Unet-All model was 125% higher than the Split dataset trained Unet-Split model. Note that the All and Split datasets had one (all text) and two (calligraphy and font) text classes, respectively. Further, Jaccard loss-trained models exhibited higher IoU metric (Table 3).

6.2 Evaluation of Image Inpainting Model

To evaluate the results of the comics image inpainting, test images I of the NoText dataset and answer masks M of the All dataset were employed. We used M to mask the part of I, to create the input images I'. The restoration I' by the image inpainting model was compared with the original images I. L1 loss, L2 loss, and PSNR were used as performance metrics. The NoText dataset trained model exhibited 2.91%p higher L1 loss compared to the HasText dataset-trained model. However, HasText-trained model exhibited 0.32%p higher L2 loss compared to the NoText-trained model (Table 4). This indicated that the image inpainting model could be trained with the text-containing comics image dataset. Since text-lacking comic images, such as the images in NoText dataset, were difficult to obtain, notably, the SZMC performance was not significantly hampered, even when the training dataset included some text-containing images.

Table 4. Evaluation of image inpainting model

Model	L1 loss	L2 loss	PSNR	Time (sec)
NoText	**8.367%**	5.506%	**74.134**	2.795
HasText	11.286%	**5.190%**	73.939	2.792

6.3 Evaluation of Whole System

To quantitatively evaluate the whole system that is combined image segmentation and inpainting, pairs of text-containing (I) and text-lacking (I') images were required. Public availability of such data is rare. Further, since multiple I' could exist due to removal of text from I, quantitative evaluation of SZMC was unattainable.

Figure 4 depicts the qualitative evaluation of SZMC. Odd and even columns depict the inputs and outputs, respectively. (a,b,c)-1 exhibited better performance, while (a,b,c)-2 recorded somewhat unstable results. Figure 4-a depicts the images with easyT that could be easily processed with the existing tools. SZMC could successfully remove easyT. Figure 4-b depicts the removal of smaller hardT (less than 75×75 pixels in a text-containing bounding box). The hardT is defined as text drawn on a complex picture and that could not be automatically processed with the existing tools. SZMC could successfully remove hardTs of

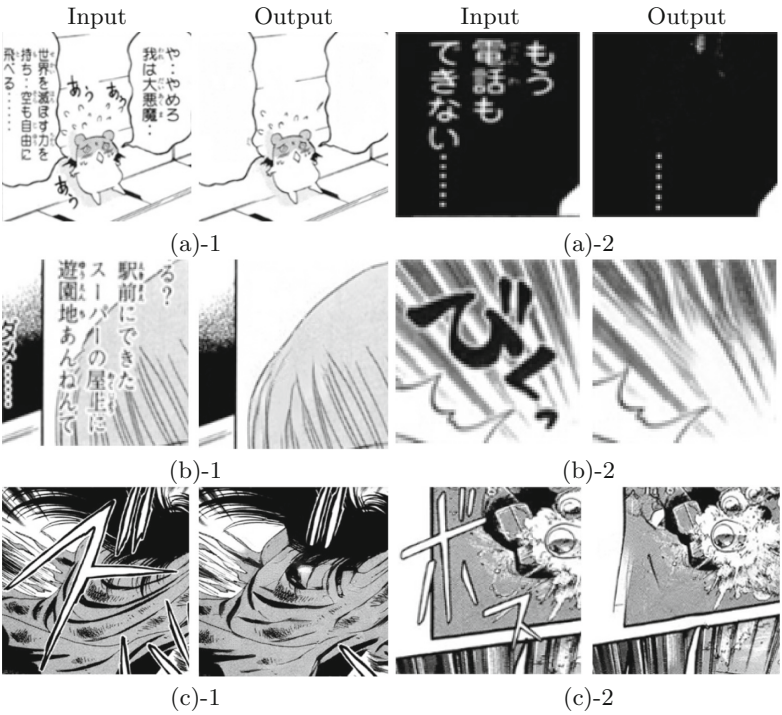

Fig. 4. Experimental results of the whole system. (a) Images that could be processed with existing tools (b) Complex images that could not be edited with existing tools (c) Images with very large calligraphic text. Image source [8], ©Arai Satoshi, ©Miyauchi Saya, ©Shimazaki Yuzuru, ©Miyone Shi

up to 75×5. Figure 4-c depicts removal of larger hardT (bounding box is larger than 125×125 pixels). SZMC exhibited inadequate image segmentation with very large hardTs (greater than 256×256 pixels). Upon adequate segmentation, larger text could be successfully removed from the image.

Figure 5 and 6 depict the experimental evaluation of the whole system. Figure 5 depicts that SZMC could process the entire comic page. Figure 6 depicts the relatively lower performance. Figure 6-a erased the non-text area due to incorrect text area segmentation. Figure 6-b depicts an effective segmentation of text areas. However, erasing was ineffective and failed to match with either the background or screen tone. This could be addressed by increasing the scale of the training dataset and optimizing the model.

7 SZMC GUI Structure

The principal users of SZMC would be professional comics editors. To aid them, we configured the publicly available SZMC in the graphic user interface (GUI) (Fig. 7). It can be downloaded from https://github.com/KUR-creative/SickZil-Machine.

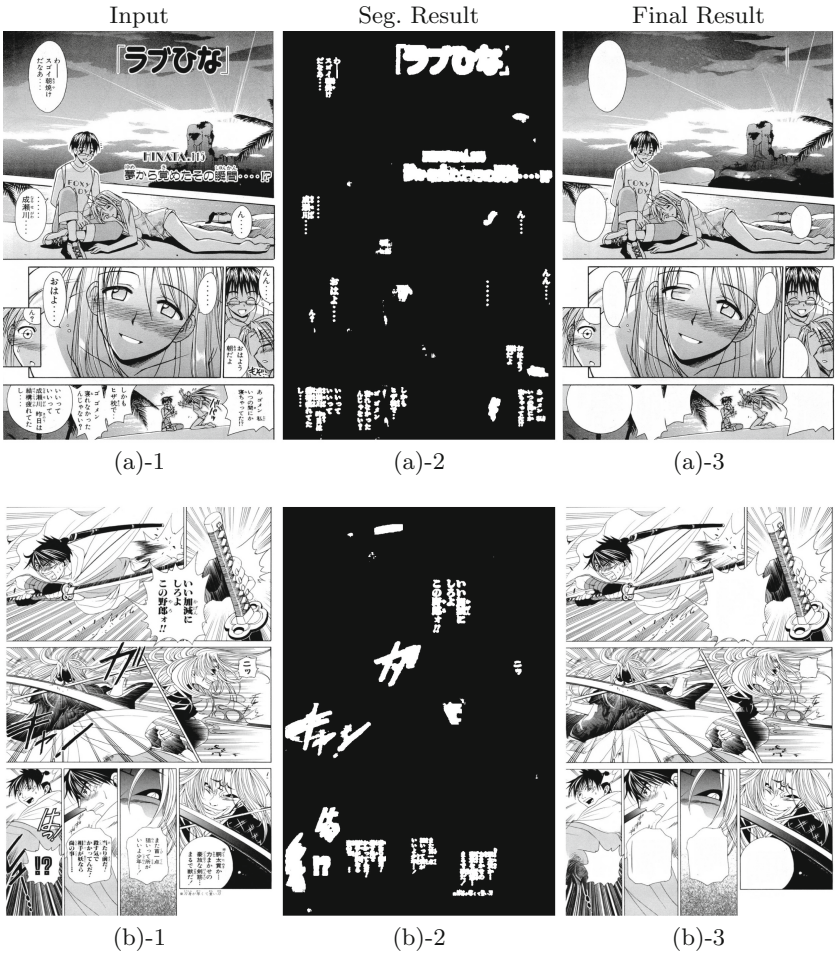

Fig. 5. Experimental results of the whole system when a complete page of comic was input. (a,b)-1 Input images, (a,b)-2 text segmentation, (a,b)-3 image inpainting. Image source: [8], ©Akamatsu Ken, ©Kobayashi Yuki

User can access the comic images directory through:[Open] – [Open Manga Project]. Further, clicking the second button on the toolbar creates the masks for all the images and according to the generated masks, automatically clears the text. Just one click of the RmTxtAll button removes texts from all the images, without any further action by the user. The segmentation masks can be selectively edited, without removing the text, by clicking the first button in the toolbar.

The third and fourth buttons create the mask and remove the text from the currently displayed image, respectively. The fifth to eighth buttons are for editing the generated masks. User can draw or erase masks, using Pen and Rectangle

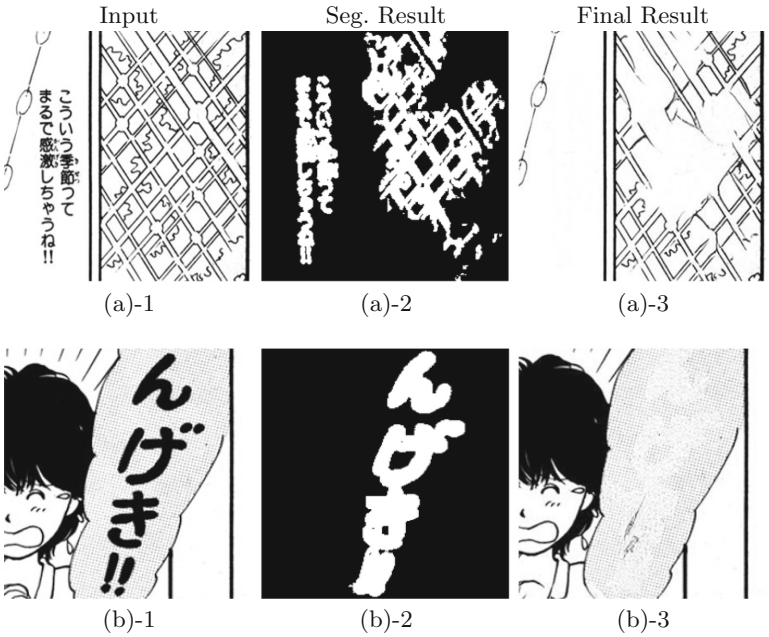

Fig. 6. Substandard cases. (a,b)-1 Input images, (a,b)-2 text segmentation, (a,b)-3 image inpainting. Image source: [8], ©Yoshi Masako

tools. The mask can be edited to either enhance text area segmentation, or to leave some text on purpose.

8 Conclusion

In this paper, we propose a two-step approach to automate the removal of the text from comic images. The SZMC segments the erasable text area from the comic images and inpainted the erased area, naturally. Therefore, the well-studied image segmentation and inpainting techniques could be applied for effective comics text removal.

We created the datasets for effective image segmentation and inpainting, using deep learning based framework, and experimentally verified the effective features. In the text segmentation datasets, grouping the calligraphic letters and font letters as one class, improved the performance. The datasets for image inpainting were not greatly affected even if the dataset had some text-containing images. Moreover, both models were quantitatively evaluated. Compared with the reference model [16], SZMC exhibited accurate image segmentation, with a slightly slower execution.

– We confirm that removing the text from comic images, which has been done manually, could be automated.

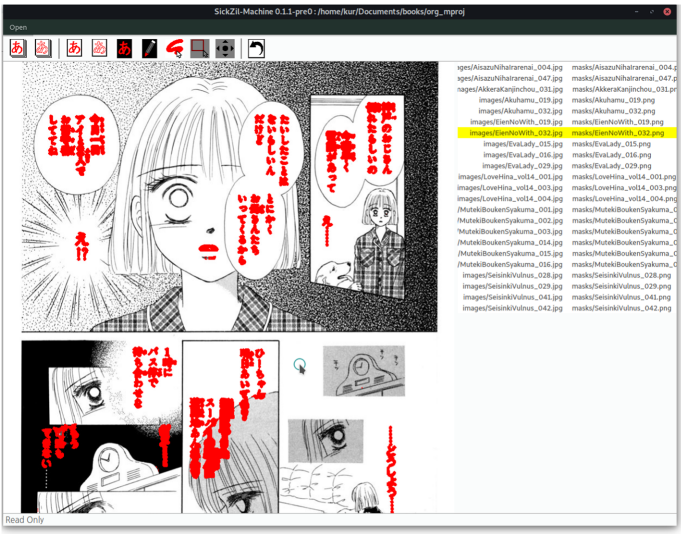

Fig. 7. Screenshot of publicly released SZMC GUI application. The screenshot exhibits the menu bar, tool bar, image edit window, and image list. Images source: [8], ©Miyauchi Saya.

- A deep learning based framework for automating comic image text removal, is presented.
- The proposed framework, SZMC, resulted in improved text area segmentation over the reference model, TSII.
- We created a dataset for removing text from comic images and identified the effective features.

In future work, we plan to release the dataset for the segmentation of the text areas in the comic images. We will create masks for all the images in Manga109 and release only the mask data. Additionally, we will explore more optimized models for better performance.

References

1. Anonymous, The Danbooru Community, Branwen, G., Gokaslan, A.: Danbooru 2018: a large-scale crowdsourced and tagged anime illustration dataset, January 2019. Accessed 1 Jan 2020
2. Aramaki, Y., Matsui, Y., Yamasaki, T., Aizawa, K.: Text detection in manga by combining connected-component-based and region-based classifications. In: 2016 IEEE International Conference on Image Processing (ICIP). IEEE, September 2016
3. Augereau, O., Iwata, M., Kise, K.: A survey of comics research in computer science. J. Imaging **4**(7), 87 (2018)
4. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Trans. Graph. **36**(4), 1–14 (2017)

5. Ito, K., Matsui, Y., Yamasaki, T., Aizawa, K.: Separation of Manga Line Drawings and Screentones, May 2015
6. Liu, G., et al.: Image inpainting for irregular holes using partial convolutions. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11215, pp. 89–105. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01252-6_6
7. Liu, X., Li, C., Wong, T.-T.: Boundary-aware texture region segmentation from manga. Comput. Vis. Med. **3**(1), 61–71 (2016). https://doi.org/10.1007/s41095-016-0069-x
8. Matsui, Y., et al.: Sketch-based manga retrieval using manga109 dataset. Multimedia Tools Appl. **76**(20), 21811–21838 (2016). https://doi.org/10.1007/s11042-016-4020-z
9. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat], November 2014
10. Nakamura, T., Zhu, A., Yanai, K., Uchida, S.: Scene Text Eraser. arXiv:1705.02772 [cs], May 2017
11. Rigaud, C., Burie, J., Ogier, J.: Segmentation-free speech text recognition for comic books. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 03, pp. 29–34, November 2017
12. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
13. Tursun, O., et al.: MTRNet: a generic scene text eraser. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, September 2019. https://doi.org/10.1109/icdar.2019.00016
14. U-ram, K., Hwan-Gue, C.: A text script removal system for comics using deep learning (Korean). In: Proceedings of Korea Computer Congress, June 2019
15. Yu, J., et al.: Free-form image inpainting with gated convolution. In: The IEEE International Conference on Computer Vision (ICCV), October 2019
16. yu45020: yu45020/Text_segmentation_image_inpainting, October 2019. https://github.com/yu45020/Text_Segmentation_Image_Inpainting. original-date: 2018–06-25T02:48:51Z
17. Zhang, S., Liu, Y., Jin, L., Huang, Y., Lai, S.: EnsNet: Ensconce Text in the Wild. arXiv:1812.00723 [cs], December 2018