Text Detection in Manga by Deep Region Proposal, Classification, and Regression

Wei-Ta Chu
Department of Computer Science and Information Engineering
Advanced Institute of Manufacturing with High-tech Innovations
National Chung Cheng University, Taiwan
Email: wtchu@ccu.edu.tw

Chih-Chi Yu
National Chung Cheng University, Taiwan
Email: john95279826@gmail.com

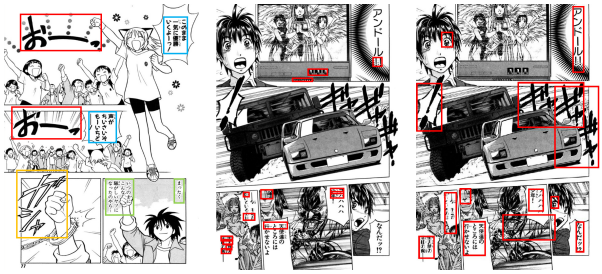
***Abstract*—Text in manga presents high variations and different contextual information, and existing scene text detection methods are not directly applicable. We propose two approaches based on deep networks to detect text in manga. In the first approach, features extracted from multiple CNNs are joined and then fed to a combination of a classification network and a regression network. In the second approach, region proposal, feature extraction, and classification/regression, are taken together in a single deep network. The evaluation results show that the first approach achieves performance comparable to the current state of the art, while the second approach yields a big performance leap over existing ones.**

I. Introduction

As more and more comic books are digitized or published digitally, the demands of efficient retrieval and management for comics become urgent in recent years. Japanese comics, i.e., manga, especially is the biggest comic industry in the world. This trend inspires the emergence of computational manga analysis [1][2]. In this work, we focus on text detection in manga, where only black and white or sometimes gray strokes are used to represent content.

Fig. 1(a) shows a manga page consisting of four types of text regions. Intuitively, the most common text region is speech balloons, which are usually with clear (white) background, as shown in the blue boxes in Fig. 1(a). If we can accurately detect text inside speech balloons, an optical character recognition module can be employed to generate text information associated with manga pages, and thus retrieval methods proposed based on text documents can be utilized. In addition to this, text is usually used to represent emotion or sound, i.e., onomatopoeia. For example, the red boxes in Fig. 1(a) represent the characters' sound, and the orange box represents the object's sound. The green boxes describe the background information of a panel. If we can detect such text regions, deeper emotion analysis or environmental settings can be discovered to facilitate automatic manga understanding. However, accurately detecting all text regions is not a trivial task due to font variation and cluttered background.

Scene text detection for natural images has been studied for years. Ones may wonder the performance of existing text detection methods for manga. Fig. 1(b) and Fig. 1(c) show sample results obtained based on one of the state-of-the-art methods (CTPN [3]) and our results, respectively. As can be seen, the CTPN method presents big problems in detecting onomatopoeia, braking sound in this case. Fig. 1(b) also shows that current text detection methods focus on detecting horizontal text lines. This observation motivates us to develop a dedicated manga text module.



(a) (b) (c)

Fig. 1. Samples of text regions (better viewed in color).

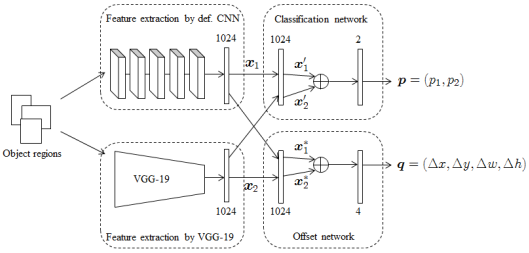


Fig. 2. Framework of the multiple CNNs approach.

In this work, we develop two deep learning approaches for manga text detection. The first method is adopting the selective search scheme to detect candidate text regions, and then verifies whether a candidate is a text region or not based on features extracted from multiple convolutional neural networks (CNNs). The second method is modified from the Faster R-CNN [4], which first detects candidates by a region proposal network, and then verifies by a CNN. We will compare these two methods with the current state of the arts.

II. Multiple CNNs Approach

Motivated by the work [5] that detects manga faces, the main steps for text detection are 1) detecting candidate text regions, 2) extracting deep features from candidates, and 3)

TABLE I
DETAILED CONFIGURATIONS OF THE NETWORKS SHOWN IN FIG. 2.

Top	Conv2D(128, 3, 3) MaxPooling2D(2, 2)	Conv2D(128, 3, 3) MaxPooling2D(2, 2)	Conv2D(256, 3, 3) MaxPooling2D(2, 2)	Conv2D(512, 3, 3) MaxPooling2D(2, 2)	Conv2D(512, 3, 3) MaxPooling2D(2, 2)	Flatten	FC(1024)
Bottom	VGG-19					Flatten	FC(1024)

verifying whether a candidate is a text region or not. In order to detect text regions in various types (c.f. Fig. 1(a)), we propose to integrate information extracted from multiple CNNs.

Fig. 2 shows the framework of multiple CNNs. Given a manga page, the selective search algorithm [6] is utilized to detect object regions. After appropriate resizing, we respectively extract features from a region by two CNNs. The top stream of Fig. 2 is a dedicated CNN containing five convolutional layers, followed by a fully-connected layer. The output of each convolutional layer is activated by the ReLU function, followed by the maxpooling process. Table I shows detailed configurations of the dedicated CNN. The notation Conv2D(k, m, n) represents that the convolution kernel is $m \times n$, and there are totally k feature maps after convolution. The notation FC(1024) represents a fully-connected (FC) layer containing 1024 nodes. The bottom stream of Fig. 2 is the VGG-19 model [7]. We take the output of the last convolutional layer, and then connect a FC layer consisting of 1024 nodes. Outputs of two streams are both embedded into 1024-dimensional vectors, denoted as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, respectively.

We develop a structure similar to [5] that contains a classification network and an offset network in the right part of Fig. 2. For classification, the vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are both fed to a FC layer, yielding $\boldsymbol{x}'_1$ and $\boldsymbol{x}'_2$, respectively. The vectors $\boldsymbol{x}'_1$ and $\boldsymbol{x}'_2$ are concatenated as $\boldsymbol{x}'$, and is then fed to a softmax layer to output a two-dimensional vector indicating the probabilities (p_1 and p_2) of a given object region being text or non-text. For estimating the offsets a region should move to more appropriately describe an object, the vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are both fed to a FC layer, yielding $\boldsymbol{x}^*_1$ and $\boldsymbol{x}^*_2$, respectively, which are then concatenated as a 2048-dimensional vector $\boldsymbol{x}^*$. The integrated vector $\boldsymbol{x}^*$ is fed to the final layer containing four nodes, outputting real values of the horizontal offset Δx, the vertical offset Δy, the width offset Δw, and the height offset Δh the given region should apply.

To train the network, two loss functions respectively derived from the classification network and the offset network are combined. The loss function L_1 from the classification network is defined as $L_1 = -\frac{1}{N}\sum(y_g \log y_p + (1-y_g)\log(1-y_p))$, where y_p is the predicted probability of the given region being text, and y_g is the ground truth where $y_g = 1$ if the given region is text, and $y_g = 0$ otherwise. The value N is the number of training data. The loss function L_2 from the offset network is defined as $L_2 = \frac{1}{N}\sum((\Delta x - \Delta\hat{x})^2 + (\Delta y - \Delta\hat{y})^2 + (\Delta w - \Delta\hat{w})^2 + (\Delta h - \Delta\hat{h})^2)$, where Δx, for example, is the truth horizontal offset, and $\Delta\hat{x}$ is the predicted horizontal offset. Two losses are then combined to be the integrated loss $L = \lambda_1 L_1 + \lambda_2 L_2$, where both weighting parameters λ_1 and λ_2 are currently set as 1. Based on the integrated loss, network parameters are determined by the Adam optimizer, with learning rate 0.001, and with mini-batch of size 30.

III. MODIFIED FASTER R-CNN

A. Network Structure

The network mentioned in Sec. II mainly focuses on extracting features from candidate regions and then classifying them into text or non-text. According to our preliminary evaluation, we found that this approach is usually limited by the performance of region proposal, i.e., the selective search algorithm. A better and essential way to tackle this problem may be designing a model that takes advantage of the characteristics of manga and then generates better region proposal for text detection.

In this work, we adopt the Faster R-CNN [4] to generate region proposals and conduct text detection in a joint framework, as shown in Fig. 3. Taking a manga page as the input, we extract features based on the ResNet model that consists of four residual blocks. Based on the derived feature maps, the region proposal network is constructed based on the idea of a fully convolutional network [8]. Particularly, a mini network with an $n \times n$ convolution kernel slides over the feature maps. This mini network takes a part of feature map as input, and maps this information into a d-dimensional vector. This vector is then fed to two sibling fully-connected layers, where one layer is for estimating how likely this part being an object (thus named as the classification layer), and another layer is for estimating the offsets this part should translate to more accurately cover an object (thus named as the regression layer).

To detect objects at multiple scales, the Faster R-CNN presents a special design called *anchor box*. Relative to the $n \times n$ *reference box* centered at (x, y), features from nine anchor boxes (covering three different scales and three different aspect ratios) nearby (x, y) are also considered. For a $W \times H$ feature map, there are thus $9WH$ anchor boxes checked by the region proposal network. The idea of anchor boxes is that the $n \times n$ reference box may not well cover an object. A $2n \times n$ anchor box, for example, may more appropriately cover an object after some resizing or translation.

Specific to manga text detection, we found that there are often text regions with extreme aspect ratios or in extremely large size, especially the onomatopoeia. Therefore, we would like to consider more aspect ratios and more size variations. In this work, anchor boxes of four different scales, and in four different aspect ratios are considered. We thus totally check 20 anchor boxes for each reference point.

Based on the region proposal network, some anchor boxes presenting high probabilities being a text region are especially taken for further verification. The text region verification task

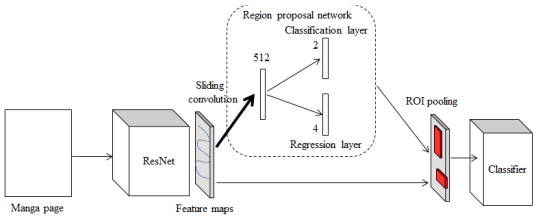


Fig. 3. Framework of the Faster R-CNN approach.

is accomplished by the Fast R-CNN [9], which takes the feature map of a region of interest (ROI), and determines whether the ROI is an object based on a sequence of convolutional layers followed by a few FC layers. By combining the region proposal network (RPN), the Faster R-CNN structure is more like a "attention-guided" Fast R-CNN.

B. Network Training

Based on an implementation pretrained on the ImageNet dataset, we construct the network shown in Fig. 3 by fine tuning the pretrained model. A two-step alternate training strategy is applied. First, we fine tune the parameters of the ResNet convolutional layers as well as the region proposal network based on the manga text training data. Second, we fine tune the parameters of the ResNet and the Fast R-CNN using the proposals detected by the RPN after fine-tuning. For the first-step fine-tuning, the loss function to be minimized is

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(\boldsymbol{t}_i, \boldsymbol{t}_i^*), \tag{1}$$

where L_{cls} denotes the classification loss, and L_{reg} denotes the regression loss. The value p_i is the predicted probability of the ith anchor box being a text region. The ground truth label p_i^* is 1 if the IoU (Intersection over Union) between the ith anchor box and the closest truth text box is larger than 0.7, and p_i^* is 0 if the IoU between the ith anchor box and the closest truth text box is less than 0.3 but larger than 0. The classification loss L_{cls} is log loss over two classes (text vs. non-text). The value $\boldsymbol{t}_i$ is a vector representing the predicted translation offsets $(\Delta x, \Delta y)$ and resizing offsets $(\Delta w, \Delta h)$, and the vector $\boldsymbol{t}_i^*$ represents the true offset vector for the ith anchor box to transform to the closest truth text box. The loss L_{reg} is defined as $R(\boldsymbol{t}_i - \boldsymbol{t}_i^*)$, where R is the robust loss function (smooth L1 norm) defined in [9]. Based on the given magna text training data, we can update network parameters by considering the characteristics of manga text, and enable the RPN outputs proposals more probably being text regions. Because many text-like anchor boxes are spatially close, we apply the non-maximum suppression scheme based on the classification score to largely reduce redundancy.

For the second-step fine-tuning, parameters of the ResNet convolutional layers and the Fast R-CNN are fine-tuned. The overall loss function is similar to that in eqn. (1). The main differences between the first step and the second step are twofold. First, in contrast to checking all anchor boxes, only the proposals raised by the RPN after the first-step tuning are used to update the Fast R-CNN. Second, when defining the ground truth label p_i^*, it is set as 1 if the IoU between the ith proposal and the closest truth text box is larger than 0.5, and p_i^* is set 0 if the IoU between the ith proposal and the closest truth text box is less than 0.5 but larger than 0.1 [9]. Network parameters are determined by the Adam optimizer, with learning rate 0.00001.

IV. Experiments

A. Evaluation Dataset

To fairly compare the proposed methods with the state of the art, we follow the same settings and use the same evaluation dataset as that in [10]. The evaluation dataset consists of six manga titles from the Manga 109 dataset [2], including DollGun (DG), Aosugiru Haru (AH), Lovehina (LH), Arisa 2 (A2), Bakuretsu KungFu Girl (BK), and Uchuka Katsuki Eva Lady (UK). We randomly select 200 manga pages from these six titles, and manually label the text regions inside. Text regions from the first 100 pages are for training, which consist of 884 text regions with clean background, and 413 text regions with cluttered background, yielding 1,297 regions in total. Text regions from the second 100 pages are for testing, which consist of 814 text regions with clean background, and 414 text regions with cluttered background.

B. Evaluation Metric

The evaluation metric follows the design in the ICDAR 2013 robust reading competition. For a detected region A and a truth text region B, we say that A corresponds to B if the ratio $r_1 = \frac{|A \bigcap B|}{|A|}$ is larger than a threshold t_p, and the ratio $r_2 = \frac{|A \bigcap B|}{|B|}$ is larger than a threshold t_r. Notice that $|\cdot|$ denotes the area of the given region. We denote the correspondence between A and B as $A \sim B$ in the following.

For a manga page that consists of a set of truth text region $\mathcal{B} = \{B_1, B_2, ..., B_k\}$, with a set of thresholds (t_p, t_r), the precision value based on the detection results $\mathcal{A} = \{A_1, A_2, ..., A_n\}$ are calculated as follows.

$$Precision = \frac{\sum_i \sum_j \delta_{ij}}{||\mathcal{A}||},\ \delta_{ij} = \begin{cases} 1 & \text{if } A_i \sim B_j \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The notation $||\mathcal{A}||$ denotes the cardinality of the set $\mathcal{A}$. The recall value is defined as

$$Recall = \frac{\sum_i \sum_j \delta_{ij}}{||\mathcal{B}||}. \tag{3}$$

Given K testing manga pages, we can calculate the average precision and recall values.

Notice that the aforementioned precision and recall values are calculated based on a specific set of thresholds t_p and t_r. With different threshold settings, the identical detection results may yield different precision and recall values. To compare different methods in a macro way, we would like to jointly consider precision and recall values with different threshold settings. To do this, we set $t_p = 0.5$ and dynamically change t_r from 0 to 1, with 0.05 stride, and calculate the corresponding precision values. For example, we denote the precision

TABLE II
PERFORMANCE COMPARISON BETWEEN DIFFERENT METHODS, IN TERMS OF P_{ov}, R_{ov}, AND F_{ov}.

	P_{ov}	R_{ov}	F_{ov}
CTPN [3]	0.16	0.14	0.15
U. Tokyo [10]	0.56	0.34	0.43
SS + Single CNN	0.50	0.36	0.42
SS + Multiple CNNs	0.49	0.51	0.50
Modified Faster R-CNN	**0.62**	**0.61**	**0.62**

value derived from $(t_p = 0.5, t_r = 0.15)$ as $P_{(0.5,0.15)}$. Conversely, we set $t_r = 0.5$ and dynamically change t_p from 0 to 1, with 0.05 stride, and calculate the corresponding precision values, e.g., $P_{(0.25,0.5)}$. We then take average of all these precision values as the *overall precision* $P_{ov} = \frac{1}{40} \times \left(\sum_j P_{(0.5,j)} + \sum_i P_{(i,0.5)}\right)$, where $i = 0, 0.05, 0.1, ..., 1$, and $j = 0, 0.05, 0.1, ..., 1$. Similarly, we can calculate the *overall recall* $R_{ov} = \frac{1}{40} \times \left(\sum_j R_{(0.5,j)} + \sum_i R_{(i,0.5)}\right)$ by considering different thresholds. To further jointly consider overall precision and overall recall, the *overall F1-measure* is calculated as $F_{ov} = \frac{2 \times P_{ov} R_{ov}}{P_{ov} + R_{ov}}$.

C. Performance Evaluation

Table II shows performance comparison between different methods. The first row shows performance obtained by the CTPN method [3]. Obviously, directly applying it to manga data is not acceptable. We implement a state-of-the-art manga text detection method proposed in [10], and achieve 0.43 overall F1-measure. The third row shows performance if only the dedicated CNN shown in Fig. 2 is used to extract features, followed by the classification network and the regression network. We see that performance comparable to [10] can be obtained. As shown in the fourth row, if features extracted from the dedicated CNN and the VGG-19 model are integrated, better performance ($F_{ov} = 0.50$) can be obtained. The last row shows that, if we replace the selective search by the region proposal network, much more performance gain can be obtained, yielding more than 0.60 in P_{ov}, R_{ov}, and F_{ov}.

Given one set of thresholds (t_p, t_r), a set of precision and recall values can be obtained. We show detailed performance variations in Fig. 4. In the top-left subfigure, for example, we fix t_r as 0.5 and change t_p from 0 to 1, and calculate the precision values corresponding to different settings. In this subfigure, we see that the multiple CNNs approach is comparable with [10] in terms of precision values, and the modified Faster R-CNN approach consistently works better. Other subfigures also show similar trends.

V. CONCLUSION

We have presented two manga text detection methods based on deep learning. In the multiple CNNs approach, region proposals are generated by the selective search algorithm, and from each candidate region features extracted from two convolutional neural networks are jointly considered. In the modified Faster R-CNN approach, text candidate regions are determined by a region proposal network, so that main processes of manga text detection, including region proposal generation, feature extraction, and classification/regression, are all modeled in a deep neural network. The evaluation results show that the modified Faster R-CNN approach makes significantly outperforming performance.

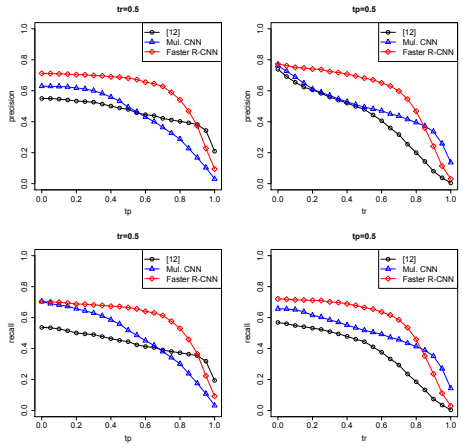


Fig. 4. Performance variations when different thresholds are set.

Acknowledgement. This work was partially supported by the Ministry of Science and Technology under the grant 107-2221-E-194-038-MY2 and MOST 107-2218-E-002-054, and the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

[1] Y. Cao, A. B. Chan, and R. W. Lau, "Automatic stylistic manga layout," in *Proceedings of SIGGRAPH Asia*, 2012.

[2] 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, 2017.

[3] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proceedings of European Conference on Computer Vision*, 2016.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] W.-T. Chu and W.-W. Li, "Manga facenet: Face detection in manga based on deep neural network," in *Proceedings of ACM International Conference on Multimedia Retrieval*, 2017, pp. 412–415.

[6] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of International Conference on Learning Representation*, 2015.

[8] J. Long, E. Shelhamer, and T. Darrelln, "Fully convolutional networks for semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[9] R. Girshick, "Fast r-cnn," in *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[10] 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 IEEE International Conference on Image Processing*, 2016.