Manga Text Detection with Manga-Specific Data Augmentation and Its Applications on Emotion Analysis

Yi-Ting Yang and Wei-Ta Chu(✉)

National Cheng Kung University, Tainan, Taiwan
wtchu@gs.ncku.edu.tw

Abstract. We especially target at detecting text in atypical font styles and in cluttered background for Japanese comics (manga). To enable the detection model to detect atypical text, we augment training data by the proposed manga-specific data augmentation. A generative adversarial network is developed to generate atypical text regions, which are then blended into manga pages to largely increase the volume and diversity of training data. We verify the importance of manga-specific data augmentation. Furthermore, with the help of manga text detection, we fuse global visual features and local text features to enable more accurate emotion analysis.

Keywords: Manga text detection · Data augmentation · Manga emotion analysis

1 Introduction

Text detection in comics is a fundamental component to facilitate text recognition and advanced comics analysis. As more and more comics are digitized and widely distributed on the internet, how to efficiently retrieve comics not only based on texture or strokes but also based on the words spoken and onomatopoeia showing sound, becomes a demanded and significant topic. To enable advanced comics understanding, a robust text detection model especially designed for comics is essential.

Although text detection for natural scene images has been widely studied for decades, directly applying them to comics does not work well [4] because the characteristics of comics is significantly different from natural images. Figure 1 shows a sample manga page consisting of different types of text. According to [2], text in manga can be categorized into four types:

- TC: Typical font type in clean background. Usually this kind of text appears in speech balloons, and is the most common type to convey main dialogue in comics. The red bounding boxes in Fig. 1 show TC text.
- AC: Atypical font type in clean background. Though in speech balloons or in clean background, text is shown in specially-designed font type to make dialogue more attractive or represent emotional content. The orange bounding boxes in Fig. 1 show AC text.

D.-T. Dang-Nguyen et al. (Eds.): MMM 2023, LNCS 13834, pp. 29–40, 2023.
https://doi.org/10.1007/978-3-031-27818-1_3

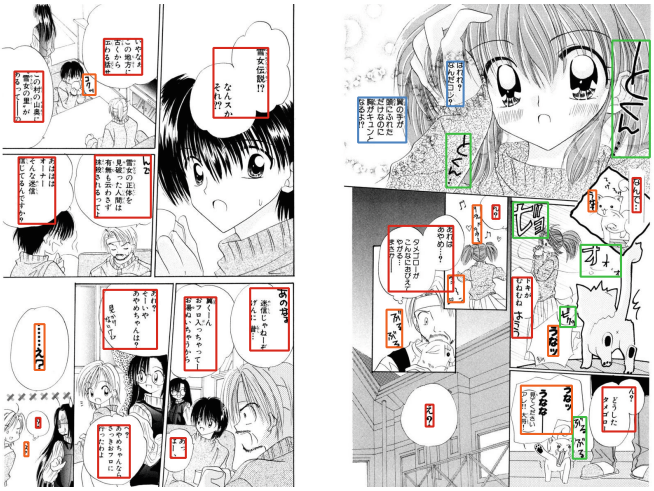

Fig. 1. A sample manga page showing different types of text. Four types of text are: TC in red, AC in orange, TD in blue, and AD in green bounding boxes, respectively. ©Ueda Miki (Color figure online)

- TD: Typical font type in dirty background. This type of text usually shows inner monologue of characters or overview of the environment. The blue bounding boxes in Fig. 1 show TD text.
- AD: Atypical font type in dirty background. This type of text is used to strengthen characters' emotion, or represent the sound made by objects or existing environmental sound. Usually onomatopoeia words like the sound of footsteps or people laughing overlay main objects or background. The green bounding boxes in Fig. 1 show AD text.

In this work, we especially focus on improving AD text detection in manga so that overall performance of text detection can be boosted. The reasons of bad detection performance for AD text detection are at least twofold. First, AD text is basically mixed with objects or background. This means features extracted from the text region are significantly "polluted". Second, as we see in the Manga109 dataset [1], AD text regions were not labeled. Even if we manually label AD text regions, the volume of training data is still far smaller than that of TC, AC, and TD. Without rich training data, discriminative features cannot be learnt to resist to the feature pollution problem.

To tackle with the aforementioned challenges, we develop a manga-specific augmentation method to increase the volume of AD text so that a better text detection model can be constructed. We develop an AD text generation model based on a generative adversarial network (GAN). With this GAN, we can generate AD text in different types at will, and augment manga pages by blending generated AD text in random sizes at random positions to largely enrich training data. We verify that, training based on the augmented data, the developed text detection module really achieves better performance.

To further verify the value of robust manga text detection, we work on manga emotion analysis by combining global visual information extracted from the entire manga page with local visual information extracted from the detected text regions. We show that, with the help of text regions, better emotion classification results can be obtained.

2 Manga Text Detection with Specific Data Augmentation

2.1 Overview

Our main objective is to generate text images in atypical styles (AD text) to significantly increase the training data, so that a stronger text detection model can be built. The so-called "text image" is an image where the spatial layout of pixels form words of a specific font type. We formulate AD text image generation as an image translation problem, as shown in Fig. 2. Given a set of text images $X = \{x_1, x_2, ..., x_m\}$ in a standard style f_s, we would like to translate them into images $\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_m\}$ in style f_t by a model, which takes X and a set of reference images $Y = \{y_1, y_2, ..., y_n\}$ in style f_t as inputs.

A text image can be divided into two parts: content and style [8] [10]. Content determines overall structure/shape of the text, and style determines finer details such as stroke width, aspect ratio, and curvature of lines. In the proposed framework, a content encoder E_c is designed to extract content representations from X, and a style encoder E_s is designed to extract style representations from Y. Content representations and style representations are jointly processed at two different levels by two sequences of residual blocks. This information is fed to a generator G to generate a text image $\hat{x}_i$ such that its content is the same as x_i and its style is similar to Y. Taking the idea of adversarial learning, a discriminator D is developed to discriminate whether the generated/translated text image $\hat{x}_i$ has similar style to the reference images Y.

2.2 Network for Augmentation

Architectures of the content encoder E_c and the style encoder E_s both follow the five convolutional layers of AlexNet [7]. We pre-train E_c and E_s before integrating them into the framework. We collect a Japanese font dataset from the FONT FREE website[1]. Totally 14 different font styles, with 142 characters in each style, are collected. Each character is represented as a text image of 227×227 pixels. These font styles are similar to that usually used to represent onomatopoeia words in manga.

For pre-training the style encoder E_s, we randomly select 113 characters from each style as the training data, and construct an AlexNet to classify each test image into one of the 14 styles. After training, the remaining 29 characters are used for testing, and the style classification accuracy is around 0.88 in our

[1] https://fontfree.me.

preliminary results. Finally, the feature extraction part (five convolutional layers) is taken as the style encoder E_s.

For pre-training the content encoder E_c, we randomly select all characters of 11 styles among the 14 styles as the training data. An AlexNet is trained to classify each test image into one of the 142 characters. After training, the characters of the remaining 3 styles are used for testing. Because the number of classes is 142, and we have very limited training data, we augment the training dataset with random erasing and random perspective processes. With data augmentation, the character classification accuracy is around 0.72 in our preliminary results. Finally, the feature extraction part (five convolutional layers) is taken as the content encoder E_c.

The final outputs of E_c and E_s are fed to a sequence of two residual blocks, as shown in Fig. 2 (dark blue lines). Motivated by [12], the content representation is processed and combined with the style representation after adaptive instance normalization (AdaIN) [5]. The main idea is to align the mean and variance of the content representations with those of style representations. After fusing two types of information, outputs of the sequence of two residual blocks (denoted as $\boldsymbol{c}'_5$) are passed to the generator G. In addition to fusing outputs of the final convolutional layers, we empirically found that fusing intermediate outputs of E_c and E_s and considering it in the generator is very helpful (light blue lines in Fig. 2, denoted as $\boldsymbol{c}'_2$). The influence of $\boldsymbol{c}'_2$ will be shown in the evaluation section.

The generator G is constructed by five convolutional layers. Inspired by [12], we think multi-level style information extracted by E_s is critical to the generation process. Different style features have different impacts in generating lines/strokes with varied curvature or widths. Denote outputs of the five convolutional layers in E_s as $\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_5$, and denote outputs of the five convolutional layers in G as $\boldsymbol{g}_1, \boldsymbol{g}_2, ..., \boldsymbol{g}_5$. Taking $\boldsymbol{c}'_5$ as the input, the generator outputs $\boldsymbol{g}_1$ by the first convolutional layer. We then concatenate $\boldsymbol{g}_1$ with $\boldsymbol{f}_4$ to be the input of the second convolutional layer. The output $\boldsymbol{g}_2$ is then concatenated with $\boldsymbol{c}'_2$ to consider multi-level style fusion as the input of the third convolutional layer. The output $\boldsymbol{g}_3$ is then concatenated with $\boldsymbol{f}_2$ to be the input of the fourth convolutional layer. After two subsequent convolutional layers, the output of the fifth convolutional layer $\boldsymbol{g}_5$ is the translated text image.

The discriminator D is also constructed by five convolutional layers. The input of the discriminator is the concatenation of the translated image $\boldsymbol{g}_5$ and one randomly-selected reference image. The goal of this discriminator is to determine whether the two input images have the same style. Inspired by PatchGAN [6], the output of the discriminator is a 14×14 map. In the map, each entry's value is between 0 and 1, and indicates how likely, in a receptive field, the translated image has the same style as the reference image. Higher style similarity between two images, higher entry value.

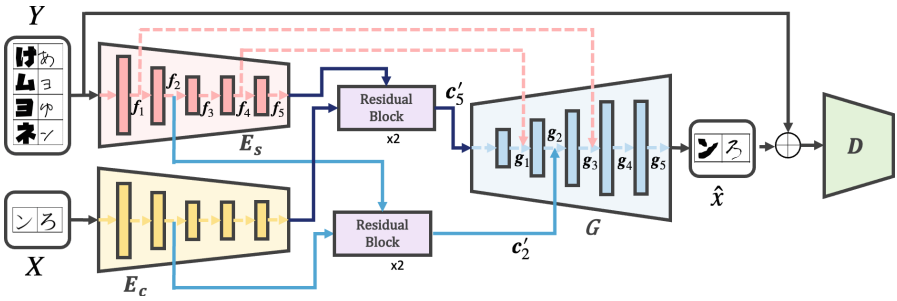

Fig. 2. Architecture of the font generation network.

2.3 Loss Functions for Augmentation

To guide network learning, we mainly rely on the adversarial loss designed for the generator G and the discriminator D:

$$\min_G \max_D \mathcal{L} = \mathbb{E}_{x\sim X, y, y_1, y_2 \sim Y}\Big[\frac{1}{196}\|D(y_1 \oplus y_2)\|_2^2 + \Big(1 - \|D\big(G(E_c(x), E_s(y)) \oplus y\big)\|_2^2\Big], \tag{1}$$

where x is an input text image from the input set X to be translated, and y is a reference image. The term $\frac{1}{196}\|D(y_1 \oplus y_2)\|_2$ is the mean L2 norm of the 14×14 map output by D. Two reference images y_1 and y_2 of the font style are randomly selected from the reference image set, and then concatenated (denoted by the operator $\oplus$). The generator G is trained to translate x into $\hat{x} = G(E_c(x), E_s(y))$ so that the discriminator D cannot distinguish the style difference between $\hat{x}$ and a randomly-selected reference image y (in the style same as y_1 and y_2).

The dataset same as training the style encoder E_s is used for training the network. With pre-trained E_c and E_s, the parameters of G and D are randomly initialized. We first freeze parameters of G, and train D for five epochs. We then alternately adjust parameters of G and D based on each mini-batch of data. The parameters of E_c and E_s are also fine-tuned in the training process. In real implementation, we adopt the Adam optimizer to adjust network parameters. The learning rate is set to 0.0002, and the momentum parameters are $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The mini-batch size is set to 4.

2.4 Augmented Manga Pages

We blend generated AD text regions into manga pages. Each text character is represented as a $W \times W$ text image. We can form a $kW \times W$ text region, denoted as Q, by concatenating k text characters, if the characters are displayed horizontally. The resolution is $W \times kW$ if the characters are displayed vertically. We then can randomly scale and rotate text regions to increase variations of augmentation results. The number k is randomly from 3 to 7.

Assume that the bounding box of the scaled and rotated region is $W' \times H'$. To blend this generated text region into a target manga page, we randomly select a region, denoted as P, of size $W' \times H'$, from this page. The selected region should not be highly-textured, and should not significantly overlap with existing text regions. Specifically, we check the ratio of the number of black pixels to the total number of the region P. The ratio for P should be less than 10%. After region selection, we blend the generated region Q into the manga page by performing exclusive OR between P and Q. For a manga page, we actually can add K randomly-generated text regions at will. This allows us to achieve different levels of augmentations.

Figure 3 shows sample augmented manga pages. As can be seen, the generated stylized text regions are seamlessly blended into manga pages. Because the positions of blending are known, we can largely increase training data to construct a model capable of detecting AD text.

Based on augmented manga pages, we train a Faster R-CNN [9] as the text detection model. We view text regions as a special type of objects, especially TD and AD text regions usually look like objects mixed with text-like texture. We adopt the ResNet-50-FPN as the backbone. This model is pre-trained based on the ImageNet dataset, and is fine-tuned based on the augmented manga dataset. Ones may wonder why we don't utilize a text detection model pre-trained based on scene text datasets, and then fine-tune it. We did it, but the experimental results don't positively support this approach. This may be because the characteristics of manga text is largely distinct from scene text.

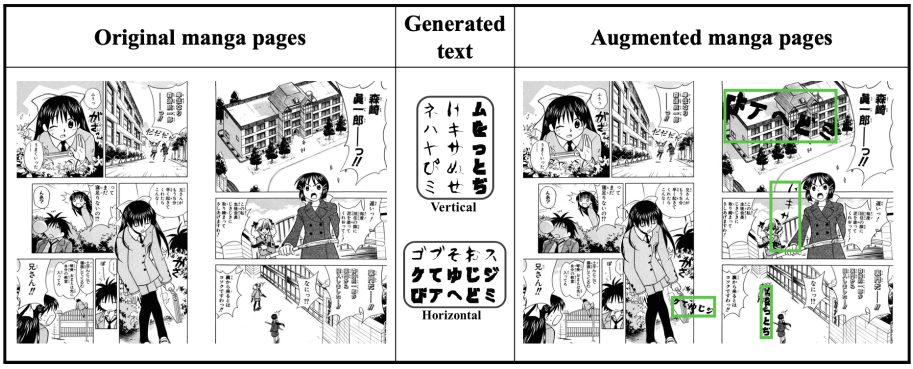

Fig. 3. Sample results of augmented manga pages. The augmented AD text regions are indicated in green boxes. ©Yagami Ken (Color figure online)

3 Manga Emotion Analysis

To verify the effectiveness of text detection on advanced manga understanding, we take emotion analysis as the exemplar application. We assume that the visual appearance of onomatopoeia words implicitly conveys emotion information, and considering this implicit information gives rise to performance gain.

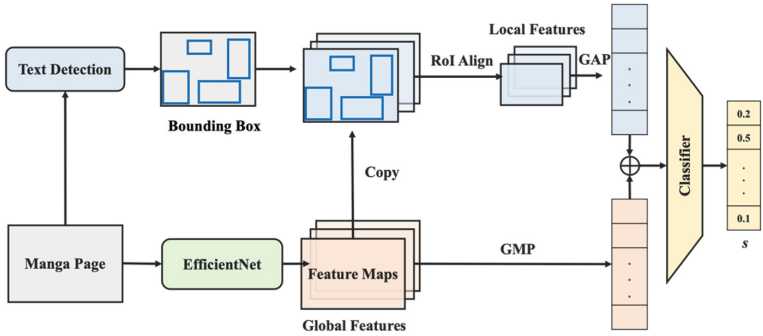

Fig. 4. Flowchart of text-assisted manga emotion recognition.

Figure 4 shows the flowchart of the proposed manga emotion recognition system. A given manga page is fed to an EfficientNet-B0 [11] to extract feature maps, which are then pooled with global maximum pooling (GMP) to represent global features $\boldsymbol{g}$ of this page. On the other hand, the proposed text detection method is applied to detect text regions in the manga page. According to text bounding boxes, we run ROIAlign [9] to get features from the detected text regions, based on the feature maps extracted by EfficientNet-B0. These features are then pooled with global average pooling (GAP) to represent local visual features $\boldsymbol{t}$ of this page. Finally, global features and local features are concatenated as $\boldsymbol{g} \bigoplus \boldsymbol{t}$, and are fed to a linear classifier with the sigmoid activation in the last layer to output a 8-dimensional real-valued vector $\boldsymbol{s}$ representing the confidence of the page showing eight different emotions.

To train the network, we calculate the asymmetric loss [3] between the predicted vector $\boldsymbol{s}$ and the ground truth vector $\boldsymbol{h}$. Given a test manga page, the proposed network outputs a 8-dimensional confidence vector $\boldsymbol{s} = (s_1, ..., s_8)$. This test page is claimed to convey the ith emotion if $s_i > 0.5$. Note that there may be multiple dimensions with values larger than the threshold 0.5.

4 Experiments of Manga Text Detection

4.1 Experimental Settings

Reference Fonts. To construct the AD text generation model, we collect Japanese font styles from the FONT FREE website. We collect totally 14 different font styles. For each font style, 142 characters are included, including Japanese hiragana and katakana. Characters in the HanaMinA style is taken as the baseline characters to be translated, i.e., the set X mentioned in Sect. 2.1. Characters in other styles are viewed as reference fonts, i.e., the set Y. This data collection is used to train the framework illustrated in Fig. 2.

The MangaAD+ Dataset. We mainly evaluate on the Manga109 dataset [1]. It is composed of 109 manga titles produced by professional manga artists in

Japan. To fairly compare our method with the state of the arts, we use a subset of Manga109 that is the same as that used in [2]. This subset consists of six manga titles, including DollGun (DG), Aosugiru Haru (AH), Lovehina (LH), Arisa 2 (A2), Bakuretsu KungFu Girl (BK) and Uchuka Katsuki Eva Lady (UK). There are totally 605 manga pages. Although the Manga109 dataset provides truth bounding boxes of text regions, most atypical text in dirty background (AD) was not labeled. To make performance evaluation more realistic and challenging, we manually label all AD regions in these six titles, and call this extensively-labeled collection MangaAD+. We randomly select 500 manga pages from this collection as the training pages, and the remaining 105 manga pages are used for testing. The numbers of TC, TD, AC, and AD regions in the 500 training pages are 5733, 626, 1635, and 1704, respectively; and the numbers of TC, TD, AC, and AD regions in the 105 testing pages are 1255, 125, 288, and 354, respectively. To augment the training data, varied numbers of generated AD regions can be blended into training pages, which are then used to fine-tune the Faster R-CNN model. Notice that we only augment the number of AD regions, rather than augmenting the number of manga pages. The testing pages (without blending generated AD text regions) are used for testing the fine-tuned models.

We follow the precision and recall values designed in the ICDAR 2013 robust reading competition, which were also used in [2] and [4]. Given the set of testing manga pages, we can calculate the average precision and recall values.

4.2 Performance Evaluation

Influence of the Number of Augmented AD Regions. In this evaluation, we control the number of augmented AD regions to augment the 500 training manga pages to different extents. The compared baselines include:

- M0: Faster R-CNN fine-tuned on the original MangaAD+ collection (without AD text augmentation).
- M1–M4: Faster R-CNN fine-tuned on the MangaAD+ collection. Each manga page is augmented with 2, 4, 6, and 8 generated AD regions, respectively.

Table 1 shows performance variations when the detection model is fine-tuned based on the MangaAD+ collection augmented at different extents. The precision, recall, and F-measure values are averaged over 105 test manga pages. We see that the overall detection performance can be effectively boosted when manga pages are augmented with generated AD regions.

Does Augmentation Based on Atypical Fonts Really Matter? What if we just blend regions of typical fonts to training manga pages? To verify this issue, we intensively augment each training page with 4 generated regions that include only typical fonts (HanaMinA style). We fine-tune the Faster R-CNN model based on this kind of augmented pages, and construct a model called M5.

Table 2 shows performance variations. Two observations can be made. First, comparing M5 with M0, fine-tuning Faster R-CNN with augmented data, even if these data are augmented with typical fonts, still has performance gain. This

Table 1. Performance variations when the Faster R-CNN model is fine-tuned based on the MangaAD+ collection augmented in different extents.

Models	Precision	Recall	F-measure
Faster RCNN-M0	0.799	0.760	0.779
Faster RCNN-M1	0.799	0.790	0.795
Faster RCNN-M2	**0.808**	**0.797**	**0.802**
Faster RCNN-M3	0.797	0.794	0.795
Faster RCNN-M4	0.789	0.788	0.789

Table 2. Performance variations when the Faster R-CNN model is fine-tuned based on the MangaAD+ collection augmented with atypical fonts and typical fonts.

Models	Precision	Recall	F-measure
Faster R-CNN-M0	0.799	0.760	0.779
Faster R-CNN-M2	**0.808**	**0.797**	**0.802**
Faster R-CNN-M5	0.791	0.784	0.787

may be because the typical fonts are blended into cluttered background, and the fine-tuned Faster R-CNN learns more from diverse data. Second, comparing M2 and M5, fine-tuning with manga-specific augmentation (M2) clearly outperforms that with common augmentation (M5). This shows that the proposed manga-specific augmentation is valuable.

Comparison with State of the Arts. We implement the best method mentioned in [2] as one of the comparison baselines. Another baseline is from the method shown in [4], which was also based on Faster R-CNN. By changing the configurations of Faster R-CNN and training based on the MangaAD+ collection, we approximate (not re-implement) the method mentioned in [4] by the implemented Faster R-CNN-M0 model.

Table 3 shows performance comparison, obtained by testing the 105 test pages in the MangaAD+ collection. As can be seen, our method significantly outperforms [2]. Although the method in [2] is also trained based on the training subset of the MangaAD+ collection, it was not designed to consider AD text, and thus misses many AD text regions.

Table 3. Performance comparison with the state of the arts.

Models	Precision	Recall	F-measure
Aramaki et al. [2]	0.638	0.351	0.453
Faster R-CNN-M0 [4]	0.799	0.760	0.779
Our (Faster R-CNN-M2)	0.808	0.797	0.802

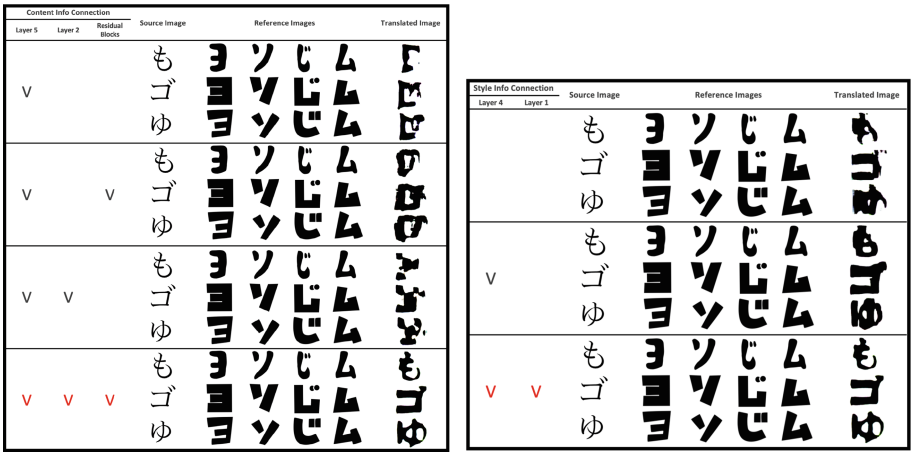

Fig. 5. Sample translated results. Left: variations when content representations from the fifth convolutional layer and the second convolution layer are considered, with or without the process of residual blocks. Right: variations when style representations from the first convolutional layer and the fourth convolutional layer are considered.

4.3 Ablation Studies

We study how the components shown in Fig. 2 influence results of translated images. The left of Fig. 5 shows translation variations when the content representations extracted from the fifth/second convolutional layers are fused with style representations, with or without the process of residual blocks. We clearly see that fusing two levels of content and style representations with residual blocks is important. The second row shows that, if only the outputs of the fifth convolutional layers are fused ($\boldsymbol{c}'_5$ mentioned in Sect. 2.2), only rough contour can be generated. If both $\boldsymbol{c}'_5$ and $\boldsymbol{c}'_2$ are considered in the generation process, better results with much finer details can be generated.

The right of Fig. 5 shows translation variations when the style representations extracted from the first/fourth convolutional layers are considered in the generation process. The first row shows that only rough content can be generated if without the skip connections of style representations. The third row shows that much better results can be obtained if both the style representations extracted from the first and the fourth convolutional layers (pink lines in Fig. 2) are considered.

5 Experiments of Manga Emotion Recognition

The MangaEmo+ Dataset. For emotion recognition, we manually label emotion classes for the 605 manga pages in the MangaAD+ dataset. Each page is labeled as a 8-dimensional binary vector, where multiple dimensions may be set as unity, showing that this page has multiple emotions.

Table 4. Performance of emotion recognition based on the MangaEmo+ collection.

Methods	micro F1	macro F1	mAP	ROC-AUC
Global only (BCE loss)	0.573	0.455	0.470	0.684
Global only (ASL)	0.650	0.541	0.511	0.726
Global+local (ASL)	0.647	0.560	0.569	0.739

Evaluation Metric. According to [13], we evaluate performance of multi-label emotion recognition in terms of micro F1, macro F1, mAP, and ROC-AUC. The micro F1 score is the harmonic mean of precision and recall rates based on the whole test samples. The macro F1 score is calculated by averaging the F1 scores corresponding to each emotion class.

Training Details. When training the network shown in Fig. 4, the SGD optimizer is adopted, with the learning rate 0.0001, weight decay 0.0005, and momentum 0.9. We set the size of a mini-batch as 4, and train the network for 50 epochs. Regarding the asymmetric loss for the MangaEmo+ collection, the positive and negative focusing parameters γ_+ and γ_- are set as 0 and 2, respectively [3]. The probability margin m is set as 0.05.

Performance Evaluation. Table 4 shows performance variations of emotion recognition based on the MangaEmo+ collection. Comparing the first two rows, when only the global features extracted by EfficientNet-B0 are considered, using ASL to guide model training brings clear performance gain over that using binary cross entropy (BCE). The ROC-AUC value boosts from 0.684 to 0.726. When global features and local visual features extracted from text regions are jointly considered, more performance gain can be obtained (ROC-AUC boosts from 0.726 to 0.739). This shows effectiveness of considering text regions in manga emotion recognition. In this case, we obtain features of text regions from the global feature maps, and concatenate local features with global features to represent the manga page. Conceptually we don't specially extract extra information from text regions. But such concatenation somehow emphasizes local visual features, and this way effectively provides performance gain.

6 Conclusion

We have presented a manga text detection method trained based on the dataset with manga-specific data augmentation. We construct a generative adversarial network to translate text images into various styles commonly used in manga. Atypical text regions are generated and are blended into manga pages to largely enrich training data. The Faster R-CNN text detection model is then fine-tuned based on this augmented dataset to achieve manga text detection. In the evaluation, we verify the effectiveness of manga-specific data augmentation, and show performance outperforming the state of the arts. We believe this is the first work targeting at atypical text detection for manga. To verify the benefit brought by

text detection in manga, we take manga emotion recognition as the exemplar application. In the future, not only manga-specific augmentation but also artist-specific augmentation can be considered. In addition, more applications related to manga understanding can be developed with the aid of detected text regions.

Acknowledgement. This work was funded in part by Qualcomm through a Taiwan University Research Collaboration Project and in part by the National Science and Technology Council, Taiwan, under grants 111-3114-8-006-002, 110-2221-E-006-127-MY3, 108-2221-E-006-227-MY3, 107-2923-E-006-009-MY3, and 110-2634-F-006-022.

References

1. Aizawa, K., et al.: Building a manga dataset "Manga109" with annotations for multimedia applications. IEEE Multimed. **27**(2), 8–18 (2020)
2. Aramaki, Y., Matsui, Y., Yamasaki, T., Aizawa, K.: Text detection in manga by combining connected-component-based and region-based classifications. In: Proceedings of IEEE ICIP, pp. 2901–2905 (2016)
3. Ben-Baruch, E., et al.: Asymmetric loss for multi-label classification. In: Proceedings of ICCV, pp. 82–91 (2021)
4. Chu, W.T., Yu, C.C.: Text detection in manga by deep region proposal, classification, and regression. In: Proceedings of IEEE VCIP, pp. 2901–2905 (2018)
5. Huang, X., Belongie, S.J.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of ICCV, pp. 1510–1519 (2017)
6. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial nets. In: Proceedings of CVPR, pp. 1125–1134 (2017)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
8. Li, W., He, Y., Qi, Y., Li, Z., Tang, Y.: FET-GAN: font and effect transfer via k-shot adaptive instance normalization. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1717–1724 (2020)
9. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems (2015)
10. Srivatsan, N., Barron, J.T., Klein, D., Berg-Kirkpatrick, T.: A deep factorization of style and structure in fonts. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (2019)
11. Tan, M., Le, Q.V.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of ICML, pp. 6105–6114 (2019)
12. Xie, Y., Chen, X., Sun, L., Lu, Y.: DG-Font: deformable generative networks for unsupervised font generation. In: Proceedings of CVPR, pp. 5130–5140 (2021)
13. Zhang, M.L., Zhou, Z.H.: A review on multi-label learning algorithms. IEEE Trans. Knowl. Data Eng. **26**(8), 1819–1837 (2014)