數位典藏與數位人文 12 期（2023 年 10 月）
Journal of Digital Archives and Digital Humanities No. 12, pp. 1-19 (October, 2023)
DOI:10.6853/DADH.202310_(12).0001

Language Model Assisted OCR Classification for Republican Chinese Newspaper Text

Konstantin Henke[1,*], Matthias Arnold[2,**]

Abstract

In this work, we present methods to obtain a neural optical character recognition (OCR) tool for article blocks in a Republican Chinese newspaper. Our basis is a small fraction of the image corpus for which text ground truth exists. We introduce a character segmentation method which produces over 90,000 labeled images of single characters and train a GoogLeNet classifier as an OCR model. In addition, we create synthetic training data from character images extracted from Song-Ti fonts. Randomly augmented on the fly and used for pre-training, they increase OCR accuracy from 95.49% to 96.95% on our test set. Finally, we employ post-OCR correction based on a pre-trained masked language model and present heuristics to select the required hyperparameters, by which we are able to correct 16% of remaining classification errors, increasing accuracy on the test set to 97.44%.

Keywords: optical character recognition, language model, ground truth, image augmentation, Republican Chinese newspapers

---

This paper is based on Konstantin Henke's Bachelor's thesis (Henke, 2021).
Manuscript received: June 24, 2022; Accepted: September 19, 2022
[1] Student, Faculty of Modern Languages, Department of Computational Linguistics, University of Heidelberg.
[2] Heidelberg Research Architecture Manager, University of Heidelberg.
[*] Email: konstantin.henke@pm.me
[**] Email: arnold@hcts.uni-heidelberg.de

1. Introduction

For more than a decade, Republican magazines and newspapers have been collected by institutes and projects now joined in the Centre for Asian and Transcultural Studies (CATS) at Heidelberg University. Our platform "Early Chinese Periodicals Online"(ECPO, https://uni-heidelberg.de/ecpo), provides open access to more than 300,000 digital images and their metadata (cf. Arnold & Hessel, 2020; Sung, Sun, & Arnold, 2014). Since the material mostly consists of image scans, the project ran a number of experiments to explore possible approaches towards full text extraction (Arnold, 2021). With regard to newspapers printed in Latin scripts, much has changed since within the last two decades. In a 2009 publication, Rose Holley still deemed the use of "'training' facility (artificial intelligence) in the OCR software" as "not viable for cost effective mass scale digitization" and noted "do not pursue" in her list of "potential methods of improving OCR accuracy" (Holley, 2009, Table 2, item 9). Today, when Liebl and Burghardt (2020) write that "transforming [historical newspapers] into machine-readable data by means of OCR poses some major challenges" they do that while they introduce their own OCR pipeline.

Unfortunately, these approaches cannot simply be adopted to historical Chinese newspapers. As we have shown in earlier works, full text extraction from these newspapers has so far been prevented especially by complex layouts and resulting difficulties to achieve reliable automatic page segmentation even within China (Arnold, 2021, 2022). In our paper we present results of an initial step towards full text extraction from a Republican China newspaper. Our basis is a small fraction of a larger image corpus for which a text ground truth has already been created. We present a character segmentation method which produces about 90,000 images of single characters. In order to expand this dataset, we generate synthetic training data. We then train a neural network as an OCR classifier and propose a method that makes use of a masked language model for further OCR error correction.

Note: We will treat single rectangular text blocks (Figure 2) as given and proceed from here to present effective methods for creating the dataset later used to train the OCR model. Due to the restricted scope of the presented experiments, this approach is still limited in terms of retrieved glyph size, image quality and font style, hence the trained model is not necessarily directly applicable to other historical Chinese documents.

2. Related Work

This section will present some of the most relevant work done in the field of document-level page segmentation, Chinese character segmentation and recognition as well as OCR post-processing. For a comprehensive overview, cf. the "Related Work" section of Henke (2021).

2.1 Document Image Segmentation

While this work assumes text blocks are already given, the question as to where to retrieve them from cannot be ignored. Eskenazi, Gomez-Krämer, and Ogier (2017) provide an exhaustive survey into the challenge of page segmentation, elaborating on both rule-based and machine learning (ML)-based approaches. In future work, we will investigate the possibility of building on the *dhSegment* tool (Oliveira, Seguin, & Kaplan, 2018).

2.2 Character Segmentation

Due to Chinese characters' nearly squared appearance, it is common to find resulting text blocks type-set and printed in a grid layout. This allows for the usage of simple image processing methods to extract single characters, such as projection profiles (Fan, Wang, & Tu., 1998; Lin, Fang, & Juang, 2001). Since however, one cannot always rely on a perfectly regular printing layout for reliable character segmentation (especially in older issues of our corpus), we aim to employ neural methods such as the *HRCenterNet* proposed by Tang, Liu, and Chiu (2020) in future work. Generally speaking, convolutional neural networks (CNNs) based on architectures like U-Net (Ronneberger, Fischer, & Brox, 2015) and YOLO (Redmon & Farhadi, 2018) are frequently employed to solve this task, but there also exist elaborate architectures tailored to specific tasks such as the recognition guided proposal network (RGPN) network proposed by Yang et al. (2018).

2.3 Image Generation and Augmentation

It is common in computer vision tasks to increase the size and diversity of the image dataset used for training by augmenting existing images and/or generating synthetic ones from scratch. This applies to Chinese OCR as well, e.g., Ren, Chen, and Sun (2016) generate single character images from 32

different fonts. Xu, Zhou, Zhang, and Fu (2018) augment images from 28 fonts with random noise, erosion and blur. Zhong, Jin, and Feng (2015) also employ non-linear transformations. For our approach, see Section 5.

2.4 Character Recognition

In Chinese character recognition, most recent work is concentrated in the field of handwritten Chinese character recognition (HCCR), cf. related work in Melnyk, You, and Li (2020)—not least due to the fact that there exist more database resources for HCCR, e.g., Liu, Yin, Wang, & Wang (2011). As for printed Chinese character recognition (PCCR), there has been a long history of non-neural approaches such as template matching techniques (Nagy, 1988). Nowadays, CNNs are the standard of addressing both HCCR and PCCR. Several pieces of work have successfully relied on the GoogLeNet architecture (Szegedy et al., 2015) for Chinese OCR (Xu et al., 2018; Yuan, Zhu, Xu, Li, & Hu, 2018; Zhong, Jin, & Xie, 2015). We follow this approach and employ a slightly modified GoogLeNet as well (see Section 6).

2.5 OCR Post-Processing

The output of context-less OCR predictions can be improved by correcting single characters based on context information. The central question is how to decide on an appropriate heuristic for (a) deciding which characters to correct (e.g., low confidence during OCR prediction) and (b) what search space to choose possibly better candidates from. Wang and Liu (2019) manually set an OCR confidence threshold of 95% (a) and have a language model choose the most likely candidate among the top five OCR candidates (b), thus combining visual similarity and language context. For our approach, see Section 7.

Finally, there do already exist tools that aim to combine all of the above into a single pipeline, cf. e.g., *Tesseract* (http://code.google.com/p/tesseract-ocr) and Ma et al. (2020).

3. The Corpus

Our corpus consists of 9,385 scanned folds from the entertainment newspaper *Jing bao* (晶報 , “The Crystal”), published 03.03.1919–23.05.1940 (Figure 1). The double-keyed text ground truth comprises all April 1939 issues

(40 folds, 762,668 characters). Aside from text blocks and their headings, it also contains mastheads, advertisements and marginalia. The methods presented below will solely focus on text blocks of uniform font-size without heading.

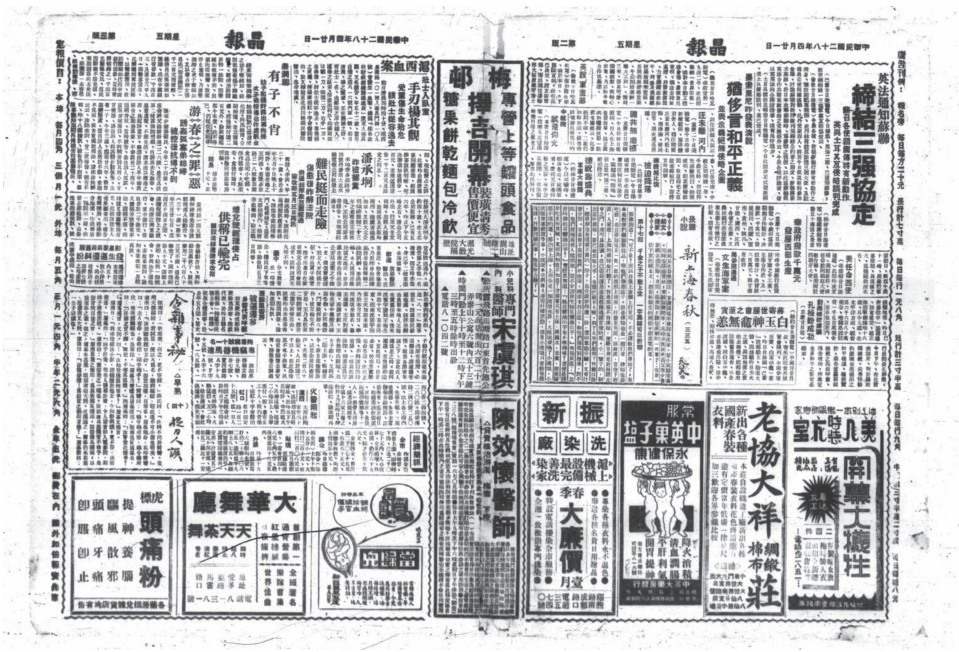


Figure 1. An example fold from *Jing bao* (晶報, "The Crystal"), April 21, 1939, pages 2–3

Source: University of Heidelberg (2023).

4. Pre-Processing and Character Segmentation

We manually crop text blocks from the April 1939 issues of the image corpus (Figure 2). Some text blocks show a certain amount of deviation from the grid layout described in Section 2.2, usually when additional characters have been squeezed into one column or because of inaccurate printing. To obtain the dataset later needed to train the classifier, we manually sort out any text blocks affected by this. Finally, we apply a rough 50-25-25 split for training, development and test set. The size of the resulting dataset can be seen in Table 1.

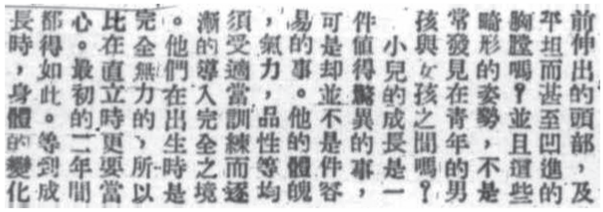


Figure 2. A manually cropped text block

Source: This study.

Table 1. The Text Block Dataset

Count	Total	Train	Dev	Test
# crops	840	426	183	231
total # chars	92039	47986	21676	22377
# unique chars	3045	2797	2074	2187

Source: This study.

Note: The dataset containing all annotated crops will be available at 10.11588/data/PVYWKB by Sept. 2022.

After adaptive binarization (kernel size: 125 px) we calculate horizontal and vertical projection profiles (cf. e.g., Fan et al., 1998). We find some text blocks are slightly rotated due to inconsistent scanning. To perform de-rotating, we find an anglee α with $\alpha \in [-2.0, -1.5, \ldots, 2.0]$ such that rotating the image by α maximizes the criterion

$$\sum_i^{w-1}(c_{i+1} - c_i)^2 + \sum_i^{h-1}(l_{i+1} - l_i)^2,$$

where w and h are the width and height of the image, c_i is the number of black pixels in the i-th column of the binary image (= the corresponding value of the vertical projection profile) and l_j in the j-th line, respectively.

In order to extract images containing single characters, we cut the gray-scale, non-binarized original text block image along separators defined by the following heuristic:

(1) Use the valleys of the vertical projection profile generated earlier to define separators between the columns.

(2) Use the valleys of the horizontal (global) projection profile to define separators between the lines.

(3) For every column, produce another (local) projection profile and the resulting separators. If a local separator lies within 7 px distance of a global separator defined above in 2., discard the global separator and only use the local separator; else only use the global separator. The result can be seen in Figure 3.

Figure 3. The text block from Fig. 2 after finding separators

Source: This study.

The positions of the valleys in the projection profiles are obtained by *scipy.signal.find_peaks* (https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html).

We further employ a normalization and contrast enhancing method using partial thresholding and linearly re-scaling grey-scale values to [0,255]. This allows even for very lightly printed characters to appear darker and have their decisive features more strongly separated from the background.

Finally, the resulting fields can be easily mapped to the ground truth text (1 field = 1 character). Empty fields caused by indentations in the text blocks (e.g., at the start of a new paragraph) have to be marked in the ground truth. The method entirely relies on correct annotation. While we can easily detect errors like missing lines (number of columns found using the projection profiles ≠ number of lines in corresponding ground truth annotation) and missing or extra characters within a line (number of fields within a specific column ≠ number of characters in the corresponding line of the ground truth), it is basically impossible for typos or swapped characters to be automatically detected. To avoid such mistakes we can only double-check annotations and manually

confirm the assigned labels in the extraction results, otherwise these errors will unavoidably lower recognition accuracy as the CNN is presented with noisy labels.

5. Character Image Generation

The method described in the section above yields a total of 92,039 character images (47,986 train, 21,676 dev, 22,377 test), but there are only 900 characters that have ≥ 10 sample images. Ren et al. (2016) argue that for PCCR, the size of the training dataset should be about two orders of magnitude larger than the number of target classes (= number of possible characters the CNN is supposed to be able to output). Consequently, we seek to generate additional synthetic training data and hypothesize that pre-training on extensive amounts of suitably augmented character images will increase the OCR accuracy for evaluation on real-life character image data and may also help to combat overfitting.

With the goal of imitating the real-life character images with artificial training data, we apply the following, partially randomized (in b., e2.2, f., g., and h) augmentations to glyph images extracted from various Song-Ti fonts (i.e. the font-style used in the newspapers) (cf. Figure 4):

(a) Extract PNG images of a predefined set of glyphs from the font file.

(b) Add random noise (peppering).

(c) Use morphological opening and then closing to enlarge noise pixels, grow them together with other close-by black pixels (other noise or the actual character) during erosion (= dilation of black contours on white background) and remove useless noise during dilation (= erosion of black pixels).

(d) Use erosion to thicken lines.

(e) Emphasize vertical lines while blurring and staining the remaining parts:

(e.1) Extract vertical elements of a certain minimum length using dilation with a vertical kernel.

(e.2) Separately apply the following:

(e.2.1) Further erode and blur the image.

(e.2.2) Generate random patches.

(e.2.3) Add the patches to the image.

(e.3) Join the result and the previously extracted vertical lines using bitwise AND.

(f) Blur the image once more. Additionally, brightness can be randomly in-/decreased before. Afterwards, linearly rescale pixel values to cover the whole 0–255 range, like the real-life images.

(g) Apply randomized elastic transformation.[1]

(h) Add padding and perform appropriate resizing.

For concrete numerical information on every augmentation step, refer to Henke (2021).

Figure 4. Augmentations applied to create synthetic training images

Source: This study.

Since, ultimately, the classes used for OCR are Unicode code points, the question arises from which code points to synthesize the additional training data. We employ the simple heuristic of using all of the glyphs featured in the ground truth that are not beyond U+9FFF, and adding any missing ones from the 4,000 most frequent characters from a representative corpus (Tsai, 2005). Furthermore, inconsistencies caused by Han-unification have to be solved. For example, the image data features 靑 instead of 青 and 淸 instead of 清 (all different code points), however only one code point exists for every other character containing 靑 / 青 as a component (請 , 情 , 靜 , ...). We decide to always use the most accurate code point as long as it's not part of the CJK Compatibility Ideographs block (U+F900...U+FAFF), so e.g., 令 (U+4EE4) is used instead of 令 (U+F9A8), even though the latter might appear more accurate, depending on the font. Generally, we find that the character variants printed in our image data to be visually closer to the Japanese standard (e.g., the components 灬 and 礻), so we choose several Japanese Song fonts for training data generation.

1 e.g., as implemented in https://gist.github.com/ chsasank/4d8f68caf01f041a6453e67fb30f8f5a

6. Character Recognition

We decide on using a GoogleNet CNN architecture (Szegedy et al., 2015), slightly modified to take 1-channel inputs instead of RGB-images. The GoogLeNet has shown to be effective in both printed and handwritten Chinese character recognition (e.g., Xu et al., 2018; Zhong, Jin, & Xie, 2015). The method described in the section above yields 4,806 classes for the final output layer. Input images are rescaled to the required input dimension of 224 × 224 px

6.1 Pre-Training

We train the CNN on the synthetic character data until the accuracy on the development set does not improve for over 50 epochs (1 epoch being equivalent to seeing all the synthetic images generated for those of the 4,806 classes that the font(s) in question provide a glyph for).

Training on different character image sets, we monitor top-k accuracies on the real-life validation set for $k \in [1, \ldots, 10]$ (cf. Table 2). While there are considerable differences between the models trained on single fonts, the most performant one is clearly the one trained on all fonts.

Table 2. Results on the Development Set

Font name	$k = 1$	2	3	5	10
Synthetic					
TW-Sung	54.40	64.86	69.51	74.18	79.30
HanaMin A	47.69	59.45	64.87	70.42	76.22
SourceHanSerif JP	62.62	70.64	74.09	77.76	81.69
I.Ming	55.60	66.41	70.90	76.09	81.23
All fonts (*)	**69.73**	**78.30**	**81.68**	**84.99**	**88.46**
Real					
Without pre-training	96.54	97.32	97.49	97.64	97.71
After pre-tr. on (*)	**97.63**	**98.57**	**98.78**	**98.98**	**99.13**

Source: This study.

Note: The bold indicate the best result in each section.

6.2 Fine-Tuning

To confirm our hypothesis, we set up further training on the 47,986 real-life character images both with pre-training on all fonts and without any pre-

training. Even including pre-training time, the former converges faster than the latter (after 5.3×10^6 vs. 7.8×10^6 seen samples), and also results in a higher development set accuracy (Table 2).

As becomes evident in Table 2, top-k accuracy significantly rises with rising k on all font combinations, even for $k = 2$, meaning that the correct prediction is often only just in second position (or at least among the top 10, for that matter).

7. LM-Based Post-OCR Correction

As explained in Section 2.5, when using a language model (LM) to post-process the Chinese character string output by the OCR model, it is not trivial to find a good heuristic for

(1) deciding which characters are likely to be wrong and hence need correction and

(2) for those characters, deciding which alternative candidates to have the language model choose from.

We address this challenge by introducing parameters t and k and proceed as follows:

(1) Let x_1 and x_2 denote the logit scores of the top two candidates output by the OCR model for a given input image. Set a threshold t. Any OCR prediction where $x_1 - x_2 < t$ is treated as likely to be incorrect (since the OCR model isn't "confident enough" its top candidate is correct) and passed to a pre-trained BERT model.[2] Test for $t \in [0, 0.5, \ldots, 10]$ to maximize accuracy.

(2) Re-predict the characters identified in (1) using the BERT model by having it choose from the top k OCR candidates. Test for $k \in [0, 1, \ldots, 18]$ to maximize accuracy.

Systematic testing yields the highest development set accuracy for $t = 2.5$ and $k = 7$.[3] For a graphical demonstration, refer to the appendix: For $k > 7$, the

2 https://huggingface.co/ckiplab/bert-base-chinese, provided by CKIP (https://ckip.iis.sinica.edu.tw) at the Academia Sinica in Taiwan.

3 In other words, the top OCR prediction is considered too unreliable if its logit score differs by less than 2.5 from the logit score of the second best candidate. The LM is used to re-predict the character in question from the top 7 OCR candidates by looking at the context. At this level, this context itself is pure OCR output, but it is assumed to be reliable enough for the LM, even though some of the context characters themselves may be corrected by the LM after.

top accuracy value (still achieved for $2.0 \leq t \leq 3.0$) slowly decreases. This is presumably due to the LM, given greater choice, becoming ever more likely to find a semantically fitting (but incorrect) candidate. Additionally, it will start to "mis-correct" characters that had been predicted correctly by the OCR model in the first place.

With $t = 2.5$ and $k = 7$ for post-processing, we attain the following final results on our test set (Table 3):

Table 3. Classification Accuracy (%)

Model	dev. set	test set
only OCR w/o pre-training	96.54	95.49
only OCR w/ pre-training	97.63	96.95
OCR w/ pre-training + LM	**98.05**	**97.44**

Source: This study.

Note: The bold indicate the best result in each section.

Hence, our post-OCR correction method additionally reduces the error by 18.1% (dev. set) / 16.1 % (test set).

8. Conclusions

Firstly, our hypothesis that pre-training on entirely synthetic character images improves the final model's performance has proven to be true in our case. The best results are achieved with pre-training on randomly augmented glyph images of as many fonts as possible (in our case, four) and fine-tuning on the segmented real-life character images. It is evident, however, that only relying on synthetic data would not suffice. This is supposedly due to the fact that its feature distribution will always be worse than that of real-life data which is naturally more similar to the dataset the model is evaluated on.

Furthermore, we have shown how during OCR post-processing, a LM is able to perform character correction to some degree. This, however, should be assumed to largely depend on whether or not enough correctly predicted context characters are given (cf. footnote 2). Hence, the use of the post-processing method presented above must be re-evaluated when working with less accurate OCR output.

9. Future Work

Our future research will build upon the promising OCR classifier, addressing bottlenecks further up the OCR pipeline. We are currently working on neural network (NN)-based character detection such as to avoid the dependency on grid-layout printing. We have identified the HRCenterNet (Tang et al., 2020) to be very suitable for this task. Further bottlenecks include:

(1) Page-level segmentation methods,

(2) Identifying headings and dealing with their different fonts and font sizes,

(3) The question if synthetic image generation can be leveraged to serve as the only source of training data whenever no suitable ground truth exists.

The outcome of addressing these issues will be decisive in the quest of providing a full OCR pipeline that produces reliable text output from scans of entire pages.

References

Arnold, M., & Hessel, L. (2020). Transforming data silos into knowledge: Early Chinese Periodicals Online (ECPO). In V. Heuveline, F. Gebhart, & N. Mohammadianbisheh (Eds.), *E-Science-Tage 2019: Data to knowledge* (pp. 95-109). Heidelberg, Germany: heiBOOKS. doi:10.11588/heibooks.598.c8420

Arnold, M. (2021). *Ground truth, neural networks, OCR: Towards full text of Republican China newspapers* [Video file]. Association for Asian Studies 2021 Virtual Annual Conference. Retrieved from https://tinyurl.com/ecpo-intro

Arnold, M. (2022). Multilingual research projects: Non-Latin script challenges for making use of standards, authority files, and character recognition. *Digital Studies/Le champ numérique*, *12*(1), 1-36. doi:10.16995/dscn.8110

Eskenazi, S., Gomez-Krämer, P., & Ogier, J.-M. (2017). A comprehensive survey of mostly textual document segmentation algorithms since 2008. *Pattern Recognition*, *64*, 1-14. doi:10.1016/j.patcog.2016.10.023

Fan, K.-C., Wang, L.-S., & Tu, Y.-T. (1998). Classification of machine-printed and handwritten texts using character block layout variance. *Pattern Recognition*, *31*(9), 1275-1284. doi:10.1016/S0031-3203(97)00143-X

Henke, K. (2021). *Building and improving an OCR classifier for Republican Chinese newspaper text* (Unpublished Bachelor's thesis). Heidelberg University, Heidelberg, Germany. doi:10.11588/heidok.00030845

Holley, R. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, *15*(3/4). doi:10.1045/march2009-holley

Liebl, B., & Burghardt, M. (2020). From historical newspapers to machine-readable data: The Origami OCR pipeline. In *Proceedings of the Workshop on Computational Humanities Research (CHR 2020)* (pp. 351-373). Retrieved from http://ceur-ws.org/Vol-2723/long20.pdf

Lin, C.-F., Fang, Y.-F., & Juang, Y.-T. (2001). Chinese text distinction and font identification by recognizing most frequently used characters. *Image and Vision Computing*, *19*(6), 329-338. doi:10.1016/S0262-8856(00)00082-2

Liu, C.-L., Yin, F., Wang, D.-H., & Wang, Q.-F. (2011). CASIA online and

offline Chinese handwriting databases. In Institute of Electrical and Electronics Engineers (Ed.), *2011 International Conference on Document Analysis and Recognition (ICDAR 2011)* (pp. 37-41). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICDAR.2011.17

Ma, W., Zhang, H., Jin, L., Wu, S., Wang, J., & Wang, Y. (2020). Joint layout analysis, character detection and recognition for historical document digitization. In Institute of Electrical and Electronics Engineers (Ed.), *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR 2020)* (pp. 31-36). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICFHR2020.2020.00017

Melnyk, P., You, Z., & Li, K. (2020). A high-performance CNN method for offline handwritten Chinese character recognition and visualization. *Soft Computing*, *24*(11), 7977-7987. doi:10.1007/s00500-019-04083-3

Nagy, G. (1988). Chinese character recognition: A twenty-five-year retrospective. In Conference on pattern recognition, IEEE Computer Society, International Association for Pattern Recognition, & Institute of Electrical and Electronics Engineers (Eds.), *9th International Conference on Pattern Recognition* (pp. 163-167). Washington, DC: IEEE Computer Society Press. doi:10.1109/ICPR.1988.28196

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv*, arXiv:1804.02767. doi:10.48550/arXiv.1804.02767

Ren, X., Chen, K., & Sun, J. (2016). A CNN based scene Chinese text recognition algorithm with synthetic data engine. *arXiv*, arXiv:1604.01891. doi:10.48550/arXiv.1604.01891

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Cham, Switzerland: Springer. doi:10.1007/978-3-319-24574-4_28

Oliveira, S. A., Seguin, B., & Kaplan, F. (2018). dhSegment: A generic deep-learning approach for document segmentation. In Institute of Electrical and Electronics Engineers (Ed.), *ICFHR 2018: 2018 16th International Conference on Frontiers in Handwriting Recognition* (pp. 7-12). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICFHR-2018.2018.00011

Sung, D., Sun, L., & Arnold, M. (2014). The birth of a database of historical periodicals: Chinese women's magazines in the late Qing and early Republican period. *Tulsa Studies in Women's Literature*, *33*(2), 227-237.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. In IEEE Computer Society (Ed.), *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1-9). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/CVPR.2015.7298594

Tang, C.-W., Liu, C.-L., & Chiu, P.-S. (2020). HRCenterNet: An anchorless approach to Chinese character segmentation in historical documents. In X. Wu, C. Jermaine, L. Xiong, X. Hu, O. Kotevska, S. Lu, ... J. Saltz (Eds.), *2020 IEEE International Conference on Big Data (Big Data)* (pp. 1924-1930). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/BigData50022.2020.9378051

Tsai, C.-H. (2005). Frequency and stroke counts of Chinese characters. Retrieved from http://technology.chtsai.org/charfreq/

University of Heidelberg . (2023). Jing bao 晶報 ("The Crystal"), April 21, 1939, pages 2–3. Early Chinese Periodicals Online. Retrieved from https://kjc-sv034.kjc.uni-heidelberg.de/ecpo/publications.php?magid=1&isid=20&ispage=2

Wang, H.-A., & Liu, P.-T. (2019). Towards a higher accuracy of optical character recognition of Chinese rare books in making use of text model. In Association for Computing Machinery (Ed.), *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage* (pp. 15-18). New York, NY: Association for Computing Machinery. doi:10.1145/3322905.3322922

Xu, X., Zhou, J., Zhang, H., & Fu, X. (2018). Chinese characters recognition from screen-rendered images using inception deep learning architecture. In B. Zeng, Q. Huang, A. El Saddik, H. Li, S. Jiang, & X. Fan (Eds.), *Advances in Multimedia Information Processing—PCM 2017* (pp. 722-732). Cham, Switzerland: Springer. doi:10.1007/978-3-319-77380-3_69

Yang, H., Jin, L., Huang, W., Yang, Z., Lai, S., & Sun, J. (2018). Dense and tight detection of Chinese characters in historical documents: Datasets and a recognition guided detector. *IEEE Access*, *6*, 30174-30183. doi:10.1109/ACCESS.2018.2840218

Yuan, T.-L., Zhu, Z., Xu, K., Li, C.-J., & Hu, S.-M. (2018). Chinese text in the

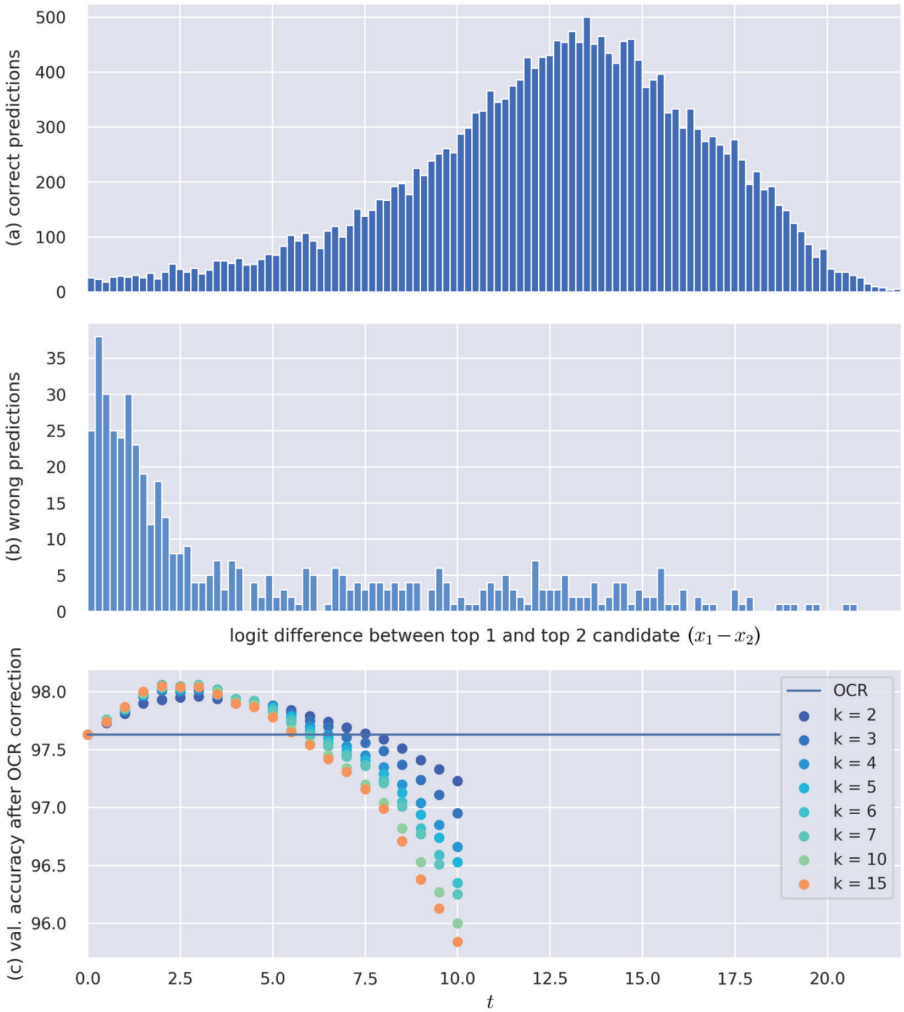
wild. *arXiv*, arXiv:1803.00085. doi:10.48550/arXiv.1803.00085

Zhong, Z., Jin, L., & Feng, Z. (2015). Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In Institute of Electrical and Electronics Engineers (Ed.), *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 96-100). Red Hook, NY: Institute of Electrical and Electronics Engineers. doi:10.1109/ICDAR.2015.7333733

Zhong, Z., Jin, L., & Xie, Z. (2015). High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps. In Institute of Electrical and Electronics Engineers (Ed.), *2015 13th International Conference on Document Analysis and Recognition (ICDAR)* (pp. 846-850). Red Hook, NY: Institute of Electrical and Electronics Engineers. doi:10.1109/ICDAR.2015.7333881

Appendix. Systematic Testing for Different Values of *t* and *k*

(a) and (b) show $x_1 - x_2$ for characters that turned out to be predicted correctly vs. incorrectly. The assumption that $x_1 - x_2$ is smaller if the top candidate is not correctly predicted is thus confirmed. (c) shows the validation accuracy (= accuracy on the development set) for various values of *t* and *k*. The original OCR accuracy of 97.63 % (i.e., without LM assistance, cf. Table 2) is marked as a horizontal line. *t* can be imagined as a vertical border separating the two bell-shaped curves in (a) and (b); the best results are achieved where this separation is optimized.



Source: This study.

數位典藏與數位人文 12 期（2023 年 10 月）
Journal of Digital Archives and Digital Humanities No. 12, pp. 1-19 (October, 2023)
DOI:10.6853/DADH.202310_(12).0001

以語言模型輔助民國報紙文本的光學字元辨識分類

Konstantin Henke[1,*], Matthias Arnold[2,**]

摘　要

本文為研發使用神經網絡的光學字元辨識（optical character recognition, OCR）工具提出了一些方法，以辨識民國時期中文報紙中的文章部分。這項工作的基礎為一小部分已存在基準真相（ground truth）的圖像語料。我們引入了一種字符分割方法，從而生成了超過 90,000 個有標籤的單一字符圖像，並且訓練了一個 GoogLeNet 分類器作為 OCR 模型。此外，我們從宋體字體中提取字符圖像，以此製作了訓練數據。這些圖像被隨機增強並被用於預訓練，測試集的 OCR 準確率由 95.49% 提高到 96.95%。最後，我們採用了基於預訓練遮罩語言模型（Masked LM）的 OCR 後校正，並提出啟發式方法來選擇所需的超參數。通過這些方法，我們能夠校正 16% 的剩餘分類錯誤，將測試集的準確率提高到 97.44%。

關鍵詞：光學字元辨識、語言模型、基準真相、圖像增強、民國時期報紙

投稿日期：2022 年 6 月 24 日；通過日期：2022 年 9 月 19 日。
[1] 海德堡大學現代語言學院計算語言學系學生。
[2] 海德堡大學海德堡研究架構經理。
[*] 通訊作者：Konstantin Henke，Email: konstantin.henke@pm.me
[**] 通訊作者：Matthias Arnold，Email: arnold@hcts.uni-heidelberg.de