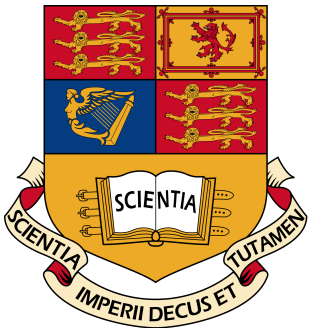


IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Natural Language Processing: Applied to Modern Literature

Author:
Michael Song

Supervisor:
Sibo Cheng

Submitted in partial fulfillment of the requirements for the MSc degree in Visual Computing and Robotics of Imperial College London

May 2023

Contents

Chapter 1

Introduction

This chapter introduces the background, challenges and main goals of this project.

1.1 Background

The collaboration between humanities and computer science has attracted increasing research attention [1]. With the development of computer technology, book digitization is being increasingly enhanced, which not only aids in preserving and spreading cultural heritage, but also the analysis of vast textual data and the extraction of knowledge in a faster, dynamic, and interactive manner [2].

Optical Character Recognition (OCR), as a way to transform images into machine-readable text data, is playing a vital role in book digitization, making it possible to analyse textual information by computers [2, 3]. Its significance extends beyond mere digitization, encompassing fields such as document management, information retrieval, and social science research [3]. With continuous advancements, OCR continues to revolutionize how we interact with printed text in the digital age.

A burgeoning area of interest in social science research is the analysis of gender discourse in early 20th-century China [4, 5]. This period witnessed profound shifts in gender dynamics, influenced by socio-political movements and ideological changes [5, 6]. Attitudes towards gender relations were undergoing a transformation, with debates emerging on women's place in society and their role in the nation's modernization [6]. Against this backdrop, Funü Zazhi [7], or "The Women's Journal", emerged as a pivotal platform for feminist discourse in China from 1915 to 1931, providing a voice for women's issues and advocating for gender equality [8].

1.2 Motivation

To analyse gender discourse in Funü Zazhi [7] using Natural Language Processing (NLP), it is necessary to transform scanned journal pages [9] into text data. However, there are several challenges in this task. Firstly, most of the Chinese characters

on the scanned images [9] are low in resolution (under 20 $\times$ 20 pixels), making them hard to distinguish. The varying image quality, considering noise, blur and distortion, adds another layer of difficulty to the task. Additionally, the text layout in Funü Zazhi [7] is inconsistent in different categories (e.g., advertisement, discussions, statement and novels), requiring the OCR method to be adaptive.

After experimenting with some ready-to-use OCR tools (e.g., PaddleOCR [10] and MacOS Preview [11]), we found their performance on the journal images [9] is poor (less than 50% accuracy in average). Therefore, it is important to develop an OCR method optimized for this task, which is also significant for digitization on relevant scanned publications.

1.3 Objectives

The main objective of this project is to apply OCR techniques to transform scanned journal pages [9] into machine-readable text data. Considering the challenges discussed in Section 1.2, it is necessary to develop an OCR method optimized for this task. The desired outcome would be a system that takes target images [9] as input and output their textual content with sensible order and high accuracy.

In addition, as the target images [9] have to be downloaded one at a time which is time-consuming (around 1000 pages in the database), a computer program is needed to download them automatically. These images should then be labeled according to their publication date and category for further analysis.

In summary, the key objectives of this project include:

1. Literature review on recent advances in OCR techniques.
2. Develop a computer program to download and label the target images [9].
3. Develop an OCR method optimized for target images.
4. Train the OCR model using synthetic data.
5. Evaluate the OCR model on target images and transfer them into text data.
6. Write the reports and presentation slides of the project.

Chapter 2

Literature Review

This chapter outlines major studies and methodologies in OCR. It will primarily focus on recent advances in the deep learning era.

2.1 Text Detection

Text detection is the task of finding text position in images, which is often considered as the first stage of OCR. The representation of text in an image can be regarded as a target, and general object detection methods are also applicable to text detection. However, text detection has different characteristics that require unique methodologies. For example, a text region can consist of a single character, a word, or multiple words. While it is not necessary to predict text labels at this stage, it poses challenges for the model to define a single text region. In this section, two types of text detection methods will be discussed.

2.1.1 Regression-Based Methods

The regression-based text detection is similar to two-category object detection: any text in the image is regarded as the same-label target, and the rest is regarded as the background. Early deep learning-based text detection algorithms were improved from object detection methods, which supports horizontal text detection.

For example, TextBoxes [12] adapts SSD [13], changing the default text box to a quadrilateral that fits various text direction and aspect ratio, providing an end-to-end solution without complex post-processing. CTPN [14] adapts Fast-RCNN [15], which extends the RPN module and designs a CRNN-based [16] module that allows the entire network to detect text sequences from convolutional features. The two-stage method obtains more accurate feature positioning through ROI Pooling.

To better handle text instances with varying sizes and orientations, TextBoxes++ [17] improves upon TextBoxes [12] by introducing multi-scale feature fusion and a new aspect ratio prediction mechanism, enhancing accuracy and robustness. EAST

[18] proposes a two-stage method for the positioning problem of tilted text, which can be trained end-to-end and can detect text in any orientation, combining simple structure and high performance.

In addition, MOST [19] introduces the TFAM module to dynamically adjust the receptive field of coarse-grained detection results, and PA-NMS to merge reliable detection prediction results based on location information. An instance-wise IoU loss function is also proposed for balanced training to handle text instances of different scales. This method can be combined with the EAST [18] to achieve better results in detecting text with extreme aspect ratios and different scales.

In summary, regression-based text detection methods leverage object detection frameworks, offering end-to-end solutions with enhanced accuracy and robustness. However, they may require post-processing and substantial computational resources.

2.1.2 Segmentation-Based Methods

Although the regression-based method has achieved good results in text detection, it is often difficult to obtain a smooth text surrounding curve for curved text, and the model is relatively complex which does not have performance advantages. Therefore, researchers have proposed text detection methods based on image segmentation. They first classify at the pixel level, determine whether each pixel belongs to a text target, obtain a probability map of the text area, and find the surrounding curve of the text through post-processing.

For example, Pixellink [20] links pixels in the same text line (word) together to segment the text. The text bounding box is extracted directly from the segmentation result without regression. However, for texts with similar positions, the text segmentation area is prone to "adhesion" problems. Wu et al. [21] proposed to segment text while learning the boundary position of the text to better distinguish text areas. Tian et al. [22] proposed to map the text pixels to a mapping space, making the vectors of same-text pixels closer and different-text pixels farther in the space.

To address the challenge in multi-scale text detection, MSR [23] suggests extracting features at various scales from an image. These features are fused, upsampled to the original size, and used to predict text center regions. The text area's outline coordinates are determined by calculating offset coordinates. To better distinguish adjacent text, PSENet [24] introduced a progressive scale expansion network. This approach learns text segmentation zones, predicts text areas with varying shrinkage ratios, and sequentially expands detected text regions. Essentially, it's a form of boundary learning that effectively tackles detecting adjacent texts of any shape. Seglink++ [25] suggests a method to represent attraction and repulsion among text block units for curved and dense text challenges. It employs a minimum spanning tree algorithm to combine units for text detection and introduces an instance-aware loss function for end-to-end training.

In conclusion, segmentation-based methods offer advantages like smooth text curve extraction and simplified models compared to regression-based approaches. How-

ever, challenges like adhesion in text segmentation and complexity may still persist.

2.2 Text Recognition

Text recognition is the task to identify text content in a fixed area. In two-stage OCR, it converts detected image patches (text region) into textual data. Recently, text recognition models leverage CNNs to transform images into feature representations. The key distinction lies in how they decode text content, primarily through either Connectionist Temporal Classification (CTC) [26] or the encoder-decoder framework [19]. In this section, we introduce recognition methods in the literature based on these two categories.

2.2.1 CTC-Based Methods

The CTC decoding method, originally used in speech recognition for sequential data, is adapted for scene text recognition by treating input images as vertical pixel frame sequences. It enables end-to-end training with only word-level annotations, eliminating the need for character-level annotations. This approach, pioneered by Graves et al. [27] in handwriting recognition, is now widely employed in scene text recognition, as seen in works by He et al. [28], Liu et al. [29], Gao et al. [30], Shi et al. [16], and Yin et al. [31].

Early endeavors in this area are exemplified by convolutional recurrent neural networks (CRNNs) [16], where RNNs are stacked atop CNNs, utilizing CTC for training and inference. DTRN [32] is a pioneering CRNN model that employs a CNN to generate convolutional feature slices, subsequently processed by RNNs. Shi et al. [16] enhance DTRN by leveraging a fully convolutional approach, exploiting the flexibility of CNNs regarding input spatial dimensions.

Gao et al. [30] diverge from the traditional RNN architecture, utilizing stacked convolutional layers to capture contextual dependencies in the input sequence with lower computational complexity and enhanced parallel computation capabilities. Yin et al. [31] innovate by simultaneously detecting and recognizing characters using character models trained end-to-end on text line images annotated with text transcripts, offering a comprehensive solution.

In general, CTC decoding enables end-to-end training with minimal annotations. While CRNNs offer effective integration of CNNs and RNNs, alternative approaches like stacked convolutional layers present computational advantages.

2.2.2 Encoder–Decoder Methods

The encoder-decoder model for sequence-to-sequence learning was initially presented by Sutskever et al. [33] for translating languages. An encoder RNN processes an input sequence and transmits its ultimate latent state to a decoder RNN, which then generates output in a self-regressive manner. This model's primary strength lies

in its ability to produce variable-length outputs, well-suited for scene text recognition. It's often paired with the attention mechanism [34] to align input and output sequences effectively. Shi et al. [35] proposed a novel end-to-end trainable neural network architecture. In this model, the CNN is employed as the feature extractor, followed by a RNN for sequence modeling, and a transcription layer to translate the RNN outputs into labels for final prediction.

Various researchers have further explored this framework. Lee and Osindero [36] presented recursive recurrent neural networks with attention modeling for lexicon-free scene text recognition. Cheng et al. [37] addressed the attention drift issue, while Bai et al. [38] introduced an edit probability (EP) metric to manage misalignment between the ground truth string and the attention's output. Liu et al. [39] proposed an efficient attention-based encoder-decoder model with a binary-constrained encoder. Besides, Shi et al. [40] proposed an arbitrary-shaped scene text recognizer based on a spatial transformer network (STN) [41], which is employed to rectify the irregular text to a regular shape before recognition. Wang et al. [42] proposed an end-to-end text spotting framework that can detect and recognize multi-oriented scene text in a single forward pass. The model consists of a shared trunk network, a text detection branch, and a text recognition branch.

Both CTC and the encoder-decoder framework streamline the recognition process, allowing training with word-level annotations. While the encoder-decoder's decoder acts as an implicit language model, incorporating linguistic knowledge, it necessitates larger training data compared to CTC, which is less language-dependent and provides better character-to-pixel alignment. Both methods, however, struggle with irregular text due to their assumption of straight text. To address this issue, Liao et al. [43] introduced a novel text recognition method, which is a segmentation-based method that treats scene text recognition as a semantic segmentation task.

2.3 End-to-end System

Text detection and recognition, traditionally treated as separate tasks, have recently been combined into end-to-end systems for extracting text from images. This shift has been facilitated by differentiable computation graphs.

Earlier methods often detected individual characters, while newer systems focus on word or line-level detection and recognition. Some systems, like those proposed by Jaderberg et al. [44] and Liao et al. [12], first detect text proposals using models like Fast-RCNN [15] or SSD [13], and then recognize the text using separate models. However, this two-step approach can lead to errors propagating between the models.

To address this, end-to-end trainable networks have emerged. These networks, such as those proposed by Bartz et al. [41] and Li et al. [45], often involve cropping feature maps instead of images and feeding them into recognition modules. Some methods use STNs [41] for word recognition or Faster-RCNN [46] for text spotting. Many recent end-to-end systems, including those by Liu et al. [39], Busta et al. [47], and He et al. [48], have similar architectures consisting of detection and recognition

branches. They often utilize methods like EAST [18] or YOLOv2 [49] for detection and incorporate CTC-based recognition modules.

Other approaches, like the one by Lyu et al. [50], modify Mask R-CNN [51] to generate character segmentation maps, while Qin et al. [52] utilize axis-aligned bounding boxes and textness segmentation masks. Notably, Xing et al. [53] introduced the first one-stage pipeline, which predicts character and text bounding boxes along with character type segmentation maps in parallel, grouping character boxes to form the final transcription.

In conclusion, the evolution of text detection and recognition has progressed significantly, with end-to-end systems offering a more streamlined approach compared to traditional two-step pipelines. While these systems have shown promising results by mitigating error propagation, challenges remain in accurately recognizing text in complex scenes. The ongoing development of one-stage pipelines suggests further potential for improving the efficiency and accuracy of end-to-end systems.

2.4 Data Synthesis

Deep learning models in text detection and recognition often face challenges due to the limited availability of labeled datasets. To address this, researchers have explored the generation of synthetic data for model training.

Jaderberg et al. [54] introduced a method for generating synthetic text recognition data by blending text with cropped natural images, applying font styles, borders, colors, and distortions. This approach proved effective, demonstrating state-of-the-art performance when trained solely on synthetic data. Gupta et al. [55] proposed SynthText, a method for embedding text into natural scenes for text detection training. By utilizing depth prediction and semantic segmentation, SynthText produced realistic images with text placed on semantically coherent surfaces. This method was instrumental in advancing text detection model performance.

Further advancements were made by Zhan et al. [56], who incorporated selective semantic segmentation and adaptive text rendering to generate more realistic synthetic text images. Their approach ensured that text appeared on sensible objects and blended with the artistic styles of the images. Liao et al. [57] introduced SynthText3D, which utilized UnrealCV to synthesize scene text images with diverse lighting, weather conditions, and occlusions. However, SynthText3D had limitations due to manual camera view selection and biased text region proposals.

Long and Yao [58] addressed these limitations with UnrealText, which featured automatic camera view generation and text region proposals based on collision detection. UnrealText significantly improved the speed and quality of synthetic text image generation, leading to better detector performance. In addition, recent research has explored text editing tasks [59, 60], aiming to replace text content while preserving styles in natural images. This approach shows potential for augmenting scene text images, although further experimentation is needed.

Chapter 3

Experiments

This chapter describes some early experiments conducted in this project.

3.1 Dataset

In this section, we introduce both the target data [9] for testing and the synthetic data for training the OCR model.

3.1.1 Target Images

The scanned images of Funü Zazhi [9] ranges from January 1915 to December 1931, where the journal is issued once a month. Each issue contains around 200 pages, so there are around 40,000 images in the database. The images are in black and white, with a dimension of roughly 750 $\times$ 1000 pixels. The text in the images is in traditional Chinese, with varying font sizes but mostly in *Song* typeface.

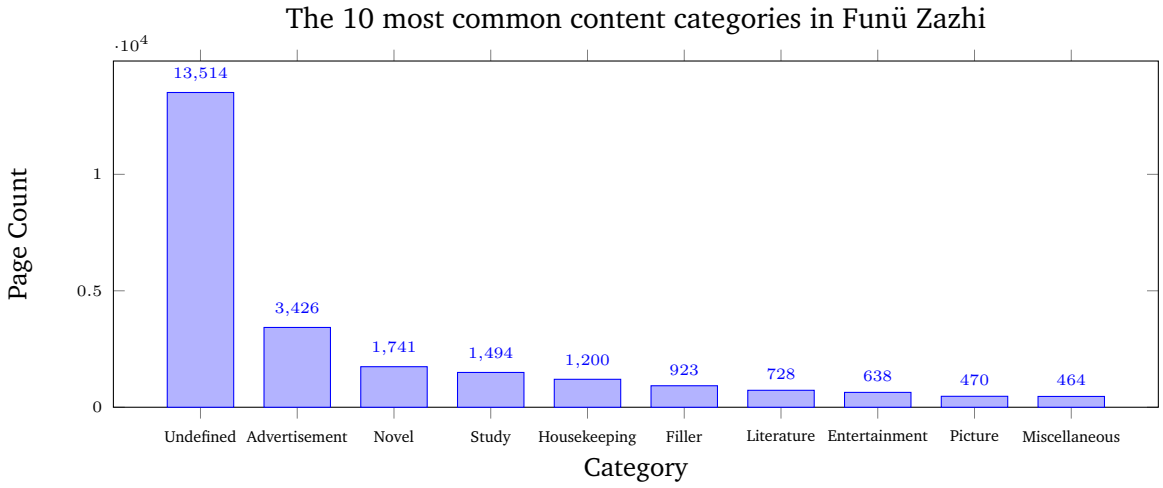
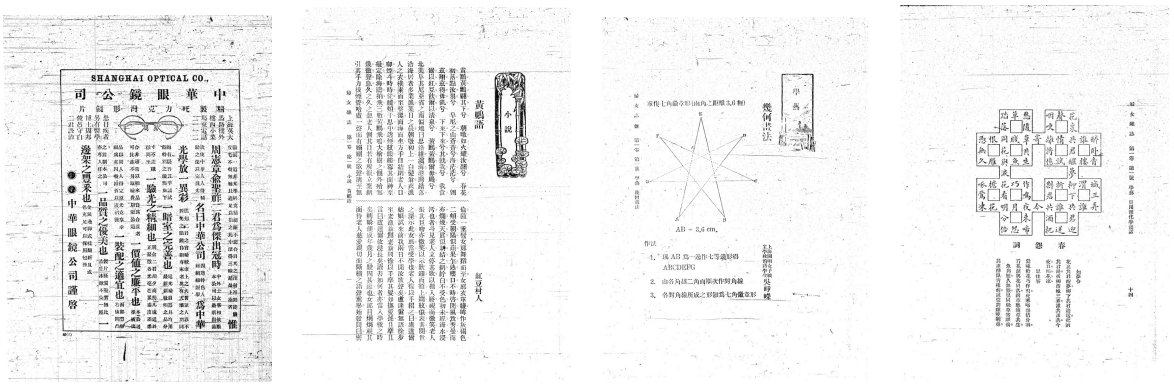


Figure 3.1: The 10 most common content categories and their page count in Funü Zazhi

There are 451 mutual exclusive content categories in Funü Zazhi, and the same for scanned images [9]. Figure 3.1 shows the 10 most common categories and their page count in the database, where most of the pages are in undefined category. The categories are manually labeled by the database administrator, which may help analyze the journal content by weighting different categories after the OCR stage.



(a) Advertisement **(b)** Novel **(c)** Study **(d)** Filler

Figure 3.2: Sample images of 4 categories

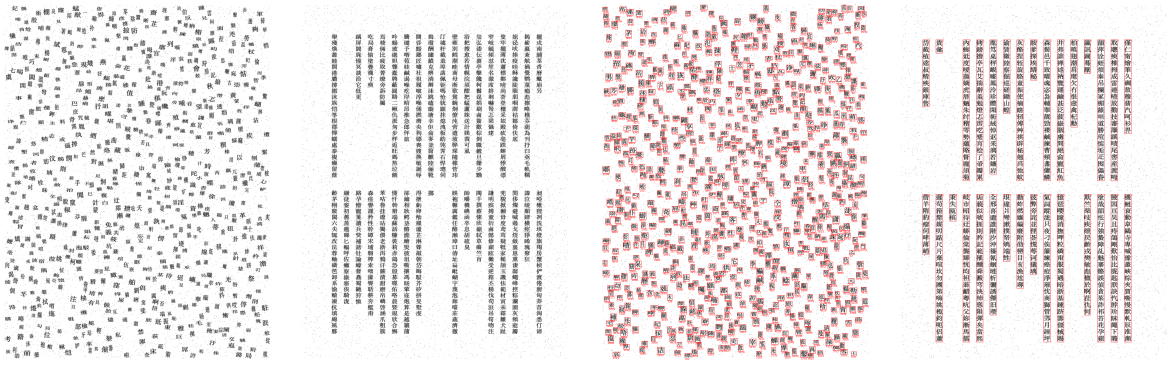
Figure 3.2 shows sample images of 4 categories in Funü Zazhi, where the text layout is different and the image quality varies, which poses challenges for the OCR task. The images are downloaded from the database [9] using a *Python* script, labeled by their issue and page number. For example, Figure 3.2a is labeled as "1501_0011.jpg", which means it is the 11th page of the January 1915 issue.

3.1.2 Synthetic Data

To train the OCR model, we use synthetic data that simulates the target images [9], as shown in Figure 3.3. The synthetic data is generated by randomly selecting characters from a list of 5,000 most common characters [61] and rendering them on a white image. To train model with different functionalities (see Section 3.2), we created two types of synthetic data:

- **Random layout** (RL): Put characters individually with random position, font sizes, aspect ratios and rotation angles, as shown in Figure 3.3a.
- **Common layout** (CL): Put characters in vertical lines with random length, position and font sizes, as shown in Figure 3.3b. This simulates the most common text layout in the target images [9].

For both types of data, random noise is added to simulate varying image quality, and the dimension is set to 750 $\times$ 1000 pixels, similar to the target images [9]. The ground truth, or expected model output of the two types is visualized in Figure 3.3c and 3.3d respectively, which consists of bounding boxes (4 integers) and labels for each character or text line. During training, the synthetic images and their ground truth are fed into the model to optimize its parameters.



(a) Random layout **(b)** Common layout **(c)** RL + ground truth **(d)** CL + ground truth

Figure 3.3: Sample synthetic images with/without ground truth

3.2 Models

In this section, we introduce the models used in early experiments, including text detection and recognition methods. We trained following models for the experiment:

- A **character classifier** (CC) based on *ResNet* [62] with a fully connected layer for 5,000 most common characters [61] in traditional Chinese.
- A **character detector** (CD) based on *Faster-RCNN* [46] with a *ResNet* backbone, which predicts bounding boxes for individual characters.
- A **line detector** (LD) based on *Faster-RCNN* [46] with a *ResNet* backbone, which predicts bounding boxes for text lines.
- A **character recognizer** (CR) based on *Faster-RCNN* [46] with a *ResNet* backbone, which predicts bounding boxes and labels for individual characters.

The experiment conducted is to compare the performance of the following methods:

- **CR**: Use the character recognizer to predict character position and label in a single forward pass.
- **CD + CC**: Use the character detector to extract image patches with a single character, and use the classifier to predict its label.
- **LD + CR**: Use the line detector to extract image patches with a text line, and use the character recognizer to predict character position/label in each line.
- **LD + CD + CC**: Use the line detector to extract image patches with a text line, then use the character detector to extract image patches with a single character in each line, and use the classifier to predict character label.

After prediction, we will reconstruct the text content according to character positions and labels, following the text layout in the image (e.g., top-to-bottom, right-to-left), although additional model is required to predict the text layout.

In the future, we will also experiment with other text recognition methods, such

as CTC-based or encoder-decoder-based models, that process the output of the line detector to predict text content. We may also consider end-to-end models that output ordered text content directly from the image.

3.3 Training

In this section, we introduce training process of the four models. The character classifier is trained using the *Adam* optimizer with a learning rate of 0.001, and the cross-entropy loss. To obtain training data, we apply data augmentation to the images of 5,000 most common [61] *Song* typeface Chinese characters, including random rotation, scaling, translation, and noise. In each epoch, 100 augmented images are generated for each character, resulting in 500,000 images in total.

The character detector and recognizer are pre-trained on the COCO dataset [63] and fine-tuned on the synthetic data RL (Section 3.1.2). In each epoch, 200 new synthetic images are generated, with 800 characters in each image. Both models are trained using the *Adam* optimizer with a learning rate of 0.0001, the smooth *L1* loss for bounding box regression, and the cross-entropy loss for character classification. The only difference is that the training data for the detector use the same label for all characters in the image, while the recognizer uses different labels for each character.

The line detector is trained similarly to the character detector, but fine-tuned on the synthetic data CL (Section 3.1.2), with 20 to 40 text lines in each image.

3.4 Evaluation

In this section, we introduce the evaluation metrics used in the experiments. For the four methods discussed in Section 3.2, we use the mean average precision (mAP) as the main evaluation metric, which is the average of precision-recall curves for each character. We also report the classification accuracy, which is the percentage of correctly predicted characters among all characters. As the experiment is not finished yet, we will share the results in the final report.

Chapter 4

Project Plan

This chapter outlines the project plan, including the timeline and potential risks.

4.1 Timeline

The planned timeline of the project is shown in Table 4.1, which is flexible and may be adjusted in the future. The project is divided into 6 stages following the objectives (Section 1.3), including literature review, data collection, model development, model training, model evaluation, and report writing. The green-labelled tasks is finished, yellow means started not finished, and red means not started. The deadline of some deliverables are 04 June for Background and Progress Report, 09 September for Final Report, and 13 September for Presentation Slides.

Objectives	29 Apr	13 May	27 May	10 Jun	23 Jun	08 Jul	22 Jul	05 Aug	19 Aug	02 Sep	16 Sep
Literature Review											
Data Collection											
Model Development											
Model Training											
Model Evaluation											
Report Writing											

Table 4.1: Planned project timeline

4.2 Risks

The potential risks of the project are listed below:

- **Data quality**: The scanned images [9] may contain noise, blur and distortion, which may affect the OCR performance.

- **Model complexity**: The OCR model may be too complex to train and evaluate, which may require more computational resources.
- **Model performance**: The OCR model may not achieve high accuracy on the target images [9], which may require further optimization.

To mitigate these risks, we plan to use data augmentation, model pruning and hyper-parameter tuning to improve the OCR performance. Transfer learning and ensemble methods may also be considered. The risks will be monitored throughout the project, and adjustments to the planned timeline will be made if necessary.

In addition, as the dataset [9] is publicly accessible and the project is for research purposes, there are no ethical issues to be concerned about. The project will be conducted in compliance with the university's ethical guidelines [64].

Bibliography

[1] Duan S, Wang J, Yang H, Su Q. Disentangling the cultural evolution of ancient China: a digital humanities perspective. Humanities and Social Sciences Communications. 2023;10(1):1-15. pages 2

[2] Tzogka C, Koidaki F, Doropoulos S, Papastergiou I, Agrafiotis E, Tiktopoulou K, et al. OCR Workflow: Facing Printed Texts of Ancient, Medieval and Modern Greek Literature. In: Qurator; 2021. . pages 2

[3] Luscombe A, Dick K, Duncan J, Walby K. Access to Information and Optical Charac-ter Recognition (OCR): A Step-by-Step Guide to Tesseract. Part One of the CAIJ Computer Literacy Series, June. 2020;(3). pages 2

[4] Bailey PJ. Gender and education in China: Gender discourses and women's schooling in the early twentieth century. Routledge; 2007. pages 2

[5] Li J. The changing discursive construction of women in Chinese popular discourse since the twentieth century. Journal of Asian Pacific Communication. 2011;21(2):238-66. pages 2

[6] Hershatter G. State of the field: Women in China's long twentieth century. The Journal of Asian Studies. 2004;63(4):991-1065. pages 2

[7] Introduction to the catalog database of Funü Zazhi;. Accessed: 2024-05-03. https://mhdb.mh.sinica.edu.tw/fnzz/. pages 2, 3

[8] Nivard J. Women and the women's press: The case of the ladies' journal (Funü Zazhi) 1915-1931. Republican China. 1984;10(1):37-55. pages 2

[9] Introduction to the catalog database of Funü Zazhi;. Accessed: 2024-05-03. https://mhdb.mh.sinica.edu.tw/fnzz/view.php. pages 2, 3, 9, 10, 13, 14

[10] PaddleOCR 2.7.3 PyPI;. Accessed: 2024-05-06. https://pypi.org/project/paddleocr/. pages 3

[11] Preview User Guide;. Accessed: 2024-05-06. https://support.apple.com/en-gb/guide/preview/welcome/mac. pages 3

[12] Liao M, Shi B, Bai X, Wang X, Liu W. Textboxes: A fast text detector with a single deep neural network. In: Proceedings of the AAAI conference on artificial intelligence. vol. 31; 2017. . pages 4, 7

[13] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer; 2016. p. 21-37. pages 4, 7

[14] Tian Z, Huang W, He T, He P, Qiao Y. Detecting text in natural image with connectionist text proposal network. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. Springer; 2016. p. 56-72. pages 4

[15] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1440-8. pages 4, 7

[16] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence. 2016;39(11):2298-304. pages 4, 6

[17] Liao M, Shi B, Bai X. Textboxes++: A single-shot oriented scene text detector. IEEE transactions on image processing. 2018;27(8):3676-90. pages 4

[18] Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, et al. East: an efficient and accurate scene text detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; 2017. p. 5551-60. pages 5, 8

[19] He M, Liao M, Yang Z, Zhong H, Tang J, Cheng W, et al. MOST: A multi-oriented scene text detector with localization refinement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2021. p. 8813-22. pages 5

[20] Deng D, Liu H, Li X, Cai D. Pixellink: Detecting scene text via instance segmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32; 2018. . pages 5

[21] Wu Y, Natarajan P. Self-organized text detection with minimal post-processing via border learning. In: proceedings of the IEEE international conference on computer vision; 2017. p. 5000-9. pages 5

[22] Tian Z, Shu M, Lyu P, Li R, Zhou C, Shen X, et al. Learning shape-aware embedding for scene text detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 4234-43. pages 5

[23] Xue C, Lu S, Zhang W. MSR: multi-scale shape regression for scene text detection. arXiv preprint arXiv:190102596. 2019. pages 5

[24] Li Y, Wu Z, Zhao S, Wu X, Kuang Y, Yan Y, et al. PSENet: Psoriasis severity evaluation network. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34; 2020. p. 800-7. pages 5

[25] Tang J, Yang Z, Wang Y, Zheng Q, Xu Y, Bai X. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. Pattern recognition. 2019;96:106954. pages 5

[26] Graves A, Fernández S, Gomez F, Schmidhuber J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning; 2006. p. 369-76. pages 6

[27] Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J. A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence. 2008;31(5):855-68. pages 6

[28] He P, Huang W, Qiao Y, Loy C, Tang X. Reading scene text in deep convolutional sequences. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30; 2016. . pages 6

[29] Liu W, Chen C, Wong KYK, Su Z, Han J. Star-net: a spatial attention residue network for scene text recognition. In: BMVC. vol. 2; 2016. p. 7. pages 6

[30] Gao Y, Chen Y, Wang J, Lu H. Reading scene text with attention convolutional sequence modeling. arXiv preprint arXiv:170904303. 2017. pages 6

[31] Yin F, Wu YC, Zhang XY, Liu CL. Scene text recognition with sliding convolutional character models. arXiv preprint arXiv:170901727. 2017. pages 6

[32] He P, Huang W, Qiao Y, Loy C, Tang X. Reading scene text in deep convolutional sequences. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30; 2016. . pages 6

[33] Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Advances in neural information processing systems. 2014;27. pages 6

[34] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014. pages 7

[35] Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo Wc. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems. 2015;28. pages 7

[36] Lee CY, Osindero S. Recursive recurrent nets with attention modeling for ocr in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2231-9. pages 7

[37] Cheng J, Zhao S, Zhang J, King I, Zhang X, Wang H. Aspect-level sentiment classification with heat (hierarchical attention) network. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management; 2017. p. 97-106. pages 7

[38] Bai F, Cheng Z, Niu Y, Pu S, Zhou S. Edit probability for scene text recognition. In: proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 1508-16. pages 7

[39] Liu Z, Li Y, Ren F, Goh WL, Yu H. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32; 2018. . pages 7

[40] Shi B, Yang M, Wang X, Lyu P, Yao C, Bai X. Aster: An attentional scene text recognizer with flexible rectification. IEEE transactions on pattern analysis and machine intelligence. 2018;41(9):2035-48. pages 7

[41] Bartz C, Yang H, Meinel C. STN-OCR: A single neural network for text detection and text recognition. arXiv preprint arXiv:170708831. 2017. pages 7

[42] Wang H, Lu P, Zhang H, Yang M, Bai X, Xu Y, et al. All you need is boundary: Toward arbitrary-shaped text spotting. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34; 2020. p. 12160-7. pages 7

[43] Liao M, Zhang J, Wan Z, Xie F, Liang J, Lyu P, et al. Scene text recognition from two-dimensional perspective. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33; 2019. p. 8714-21. pages 7

[44] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Reading text in the wild with convolutional neural networks. International journal of computer vision. 2016;116:1-20. pages 7

[45] Li H, Wang P, Shen C. Towards end-to-end text spotting with convolutional recurrent neural networks. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 5238-46. pages 7

[46] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;28. pages 7, 11

[47] Busta M, Neumann L, Matas J. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2204-12. pages 7

[48] He T, Tian Z, Huang W, Shen C, Qiao Y, Sun C. An end-to-end textspotter with explicit alignment and attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 5020-9. pages 7

[49] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger; 2016. pages 8

[50] Lyu P, Liao M, Yao C, Wu W, Bai X. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 67-83. pages 8

[51] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision; 2017. p. 2961-9. pages 8

[52] Qin S, Bissacco A, Raptis M, Fujii Y, Xiao Y. Towards unconstrained end-to-end text spotting. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 4704-14. pages 8

[53] Xing L, Tian Z, Huang W, Scott MR. Convolutional character networks. In: Proceedings of the IEEE/CVF international conference on computer vision; 2019. p. 9126-36. pages 8

[54] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:14062227. 2014. pages 8

[55] Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2315-24. pages 8

[56] Zhan F, Lu S, Xue C. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018. p. 249-66. pages 8

[57] Liao M, Song B, Long S, He M, Yao C, Bai X. SynthText3D: synthesizing scene text images from 3D virtual worlds. Science China Information Sciences. 2020;63:1-14. pages 8

[58] Long S, Yao C. Unrealtext: Synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:200310608. 2020. pages 8

[59] Wu L, Zhang C, Liu J, Han J, Liu J, Ding E, et al. Editing text in the wild. In: Proceedings of the 27th ACM international conference on multimedia; 2019. p. 1500-8. pages 8

[60] Yang Q, Huang J, Lin W. Swaptext: Image based texts transfer in scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 14700-9. pages 8

[61] Chinese Text Computing;. Accessed: 2024-05-03. `https://lingua.mtsu.edu/chinese-computing/statistics/`. pages 10, 11, 12

[62] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition; 2015. pages 11

[63] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer; 2014. p. 740-55. pages 12

[64] Legal, Social, Ethical and Professional Requirements;. Accessed: 2024-05-16. https://imperialcollege.atlassian.net/wiki/spaces/docteaching/pages/152420345/Legal+Social+Ethical+and+Professional+Requirements. pages 14