Chinese Character Recognition: Applied to Modern Literature

Author: Michael Song
Supervisor: Sibo Cheng
Second Marker: Pancham Shukla

Imperial College London

September 8, 2024

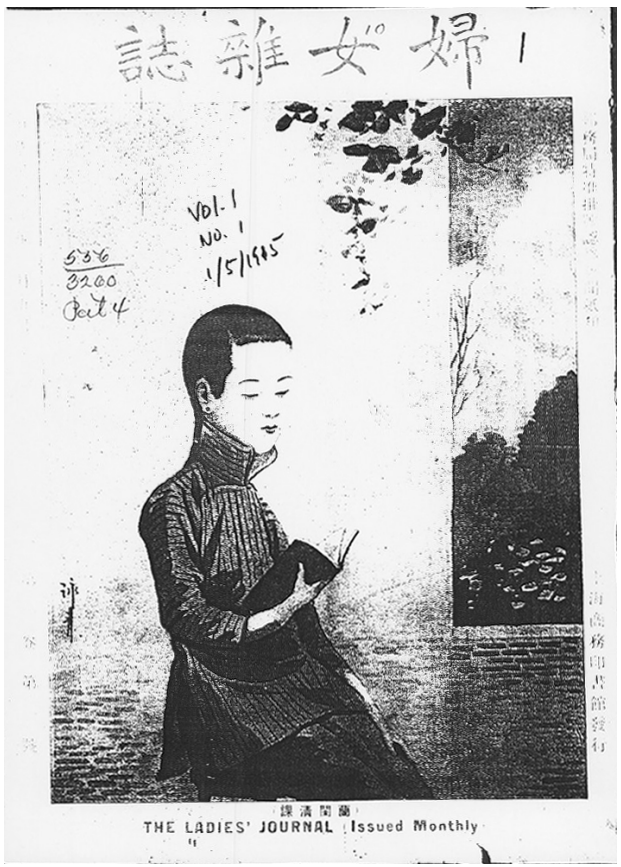
Background

- Optical Character Recognition (OCR)[1]: converting images of text into machine-encoded text.

[1] Image source: click here

Background

- Funü Zazhi[2]: a historical Chinese feminist journal (1915-1931), valuable for research on gender discourse.



[2]Image source: click here

Motivation

- We want to analyze the content of Funü Zazhi by Natural Language Processing (NLP) using **computers**.
- However, the digital database[3] of Funü Zazhi only contains scanned **images**.
- Therefore, we need to convert these images into machine-readable **text**!

[3] https://mhdb.mh.sinica.edu.tw/fnzz/view.php

Challenges

- Low resolution characters, diverse text layouts, varying image quality...

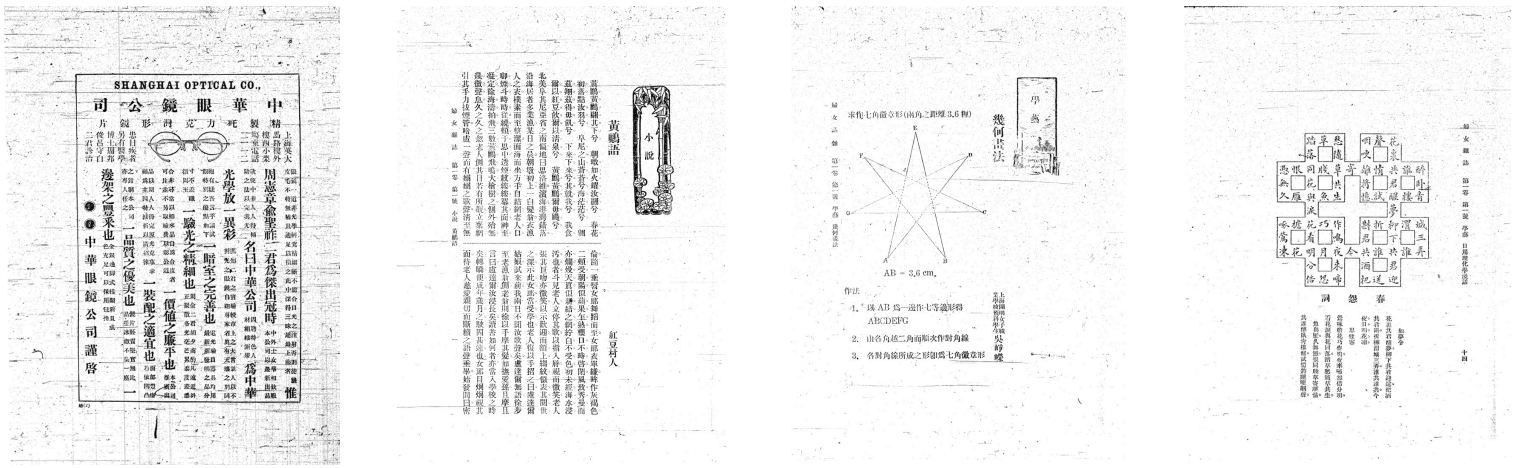


Figure: Sample images of Funü Zazhi[4]

[4] Image source: click here

Challenges

- Lack of annotated datasets with **similar features**.
- OCR tools trained on common datasets have poor performance on Funü Zazhi (less than 50% accuracy!).

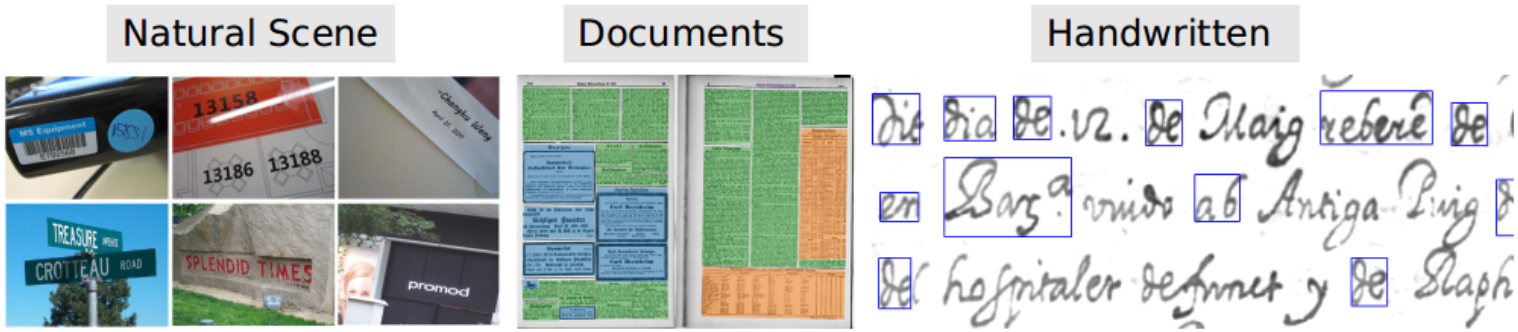


Figure: Common OCR datasets that do not resemble Funü Zazhi[5]

[5] Image source: click here

Main Goals

- Develop an optimized OCR system.
- Generate annotated synthetic data to simulate Funü Zazhi images for training.
- Perform OCR on 36,101 Funü Zazhi images.

Related Work

- Text Detection: locate text regions in the image.
- Text Recognition: convert text region into text.
- Two-stage OCR: separate detection and recognition into two models.
- End-to-end OCR: combine detection and recognition into a single model.

Our Approach

- Two-stage OCR system.
- Special text ordering module for Funü Zazhi.

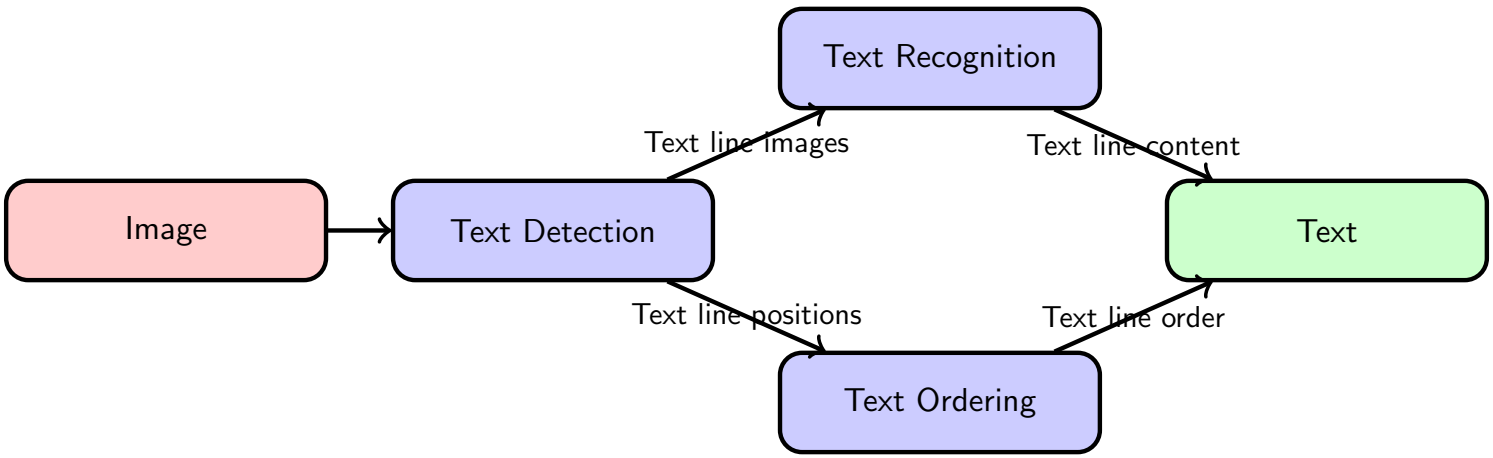


Figure: Overall structure of the OCR system for Funü Zazhi

Text Detection

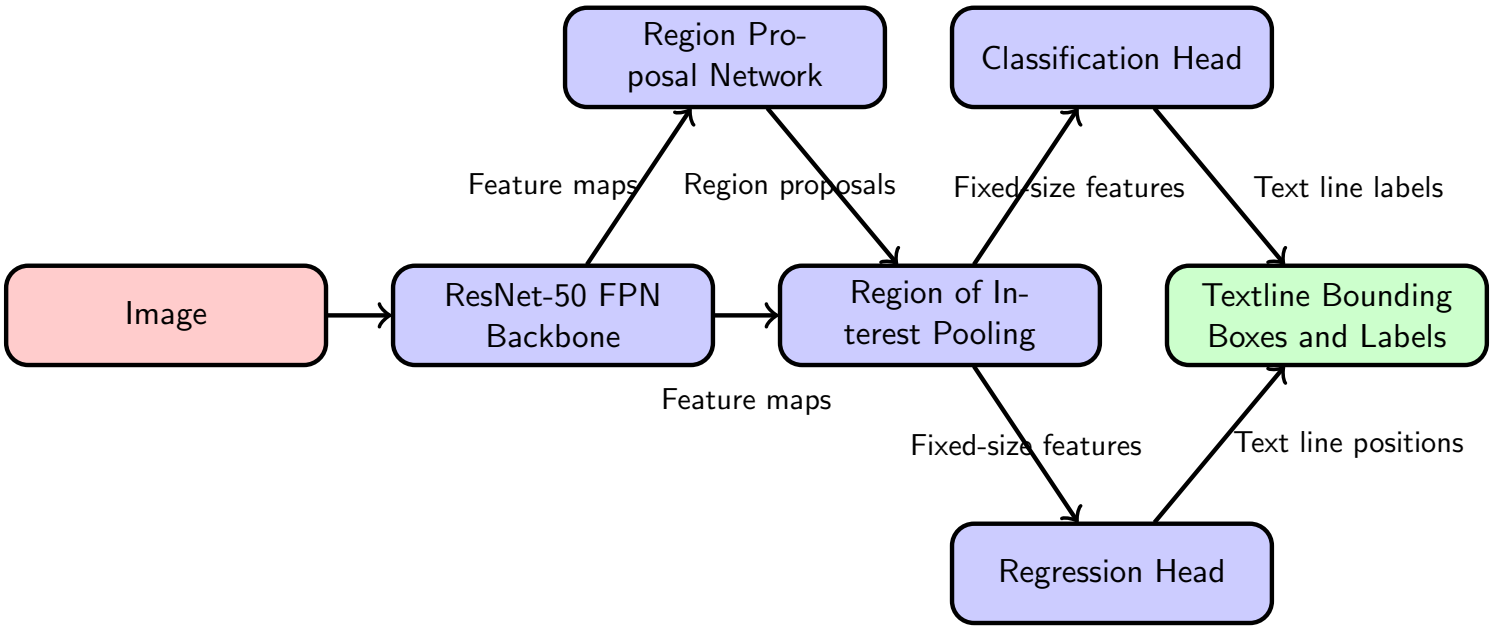


Figure: Faster R-CNN[6] architecture for text detection

[6] https://arxiv.org/pdf/1506.01497

Text Recognition

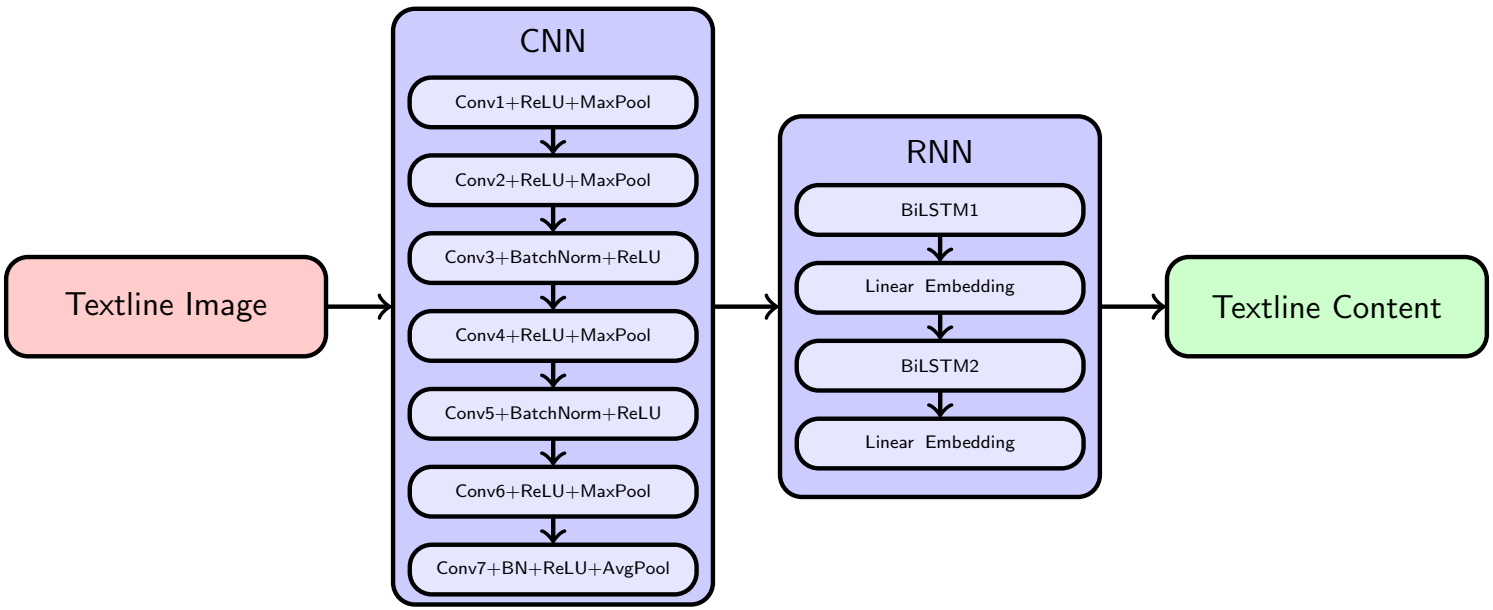


Figure: CRNN[7] architecture for text recognition

[7] https://arxiv.org/pdf/1507.05717

Text Ordering

- By default, the detected text lines are ordered by their confidence scores, not the actual **reading order**.
- We need to reorder the text lines to make the whole text readable.
- The text ordering module works based on text line **positions** and **labels** (vertical/horizontal) given by the text detection model.

Text Ordering

- Around 80% of the Funü Zazhi images are in a specific layout, we call it L1.
- Other images have different layouts, we call them L2.
- The algorithm first distinguishes between L1 and L2, by counting the number of **vertical** text lines in the upper and lower parts of the image respectively.

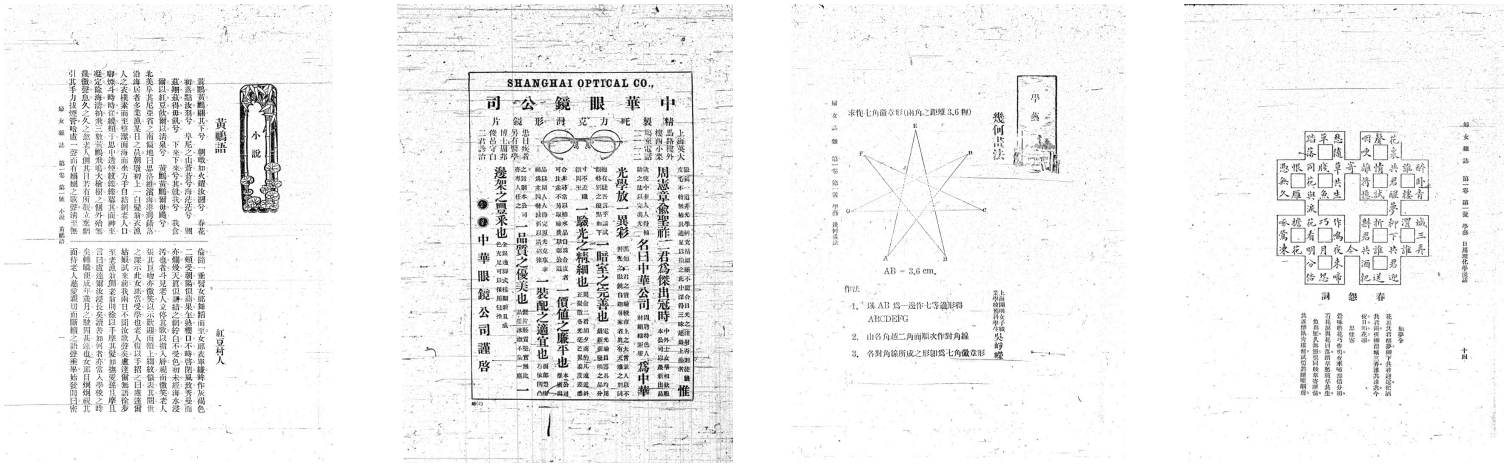
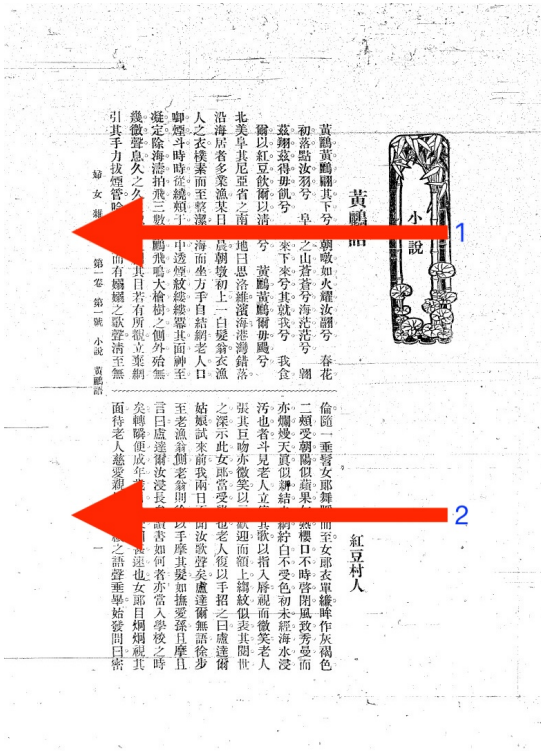


Figure: We regard the left image as L1 and other images as L2

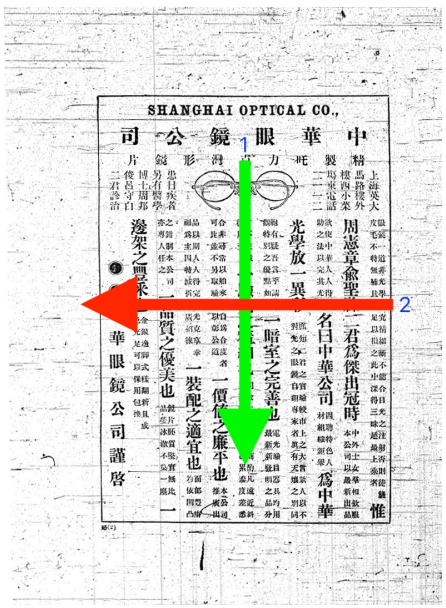
Text Ordering

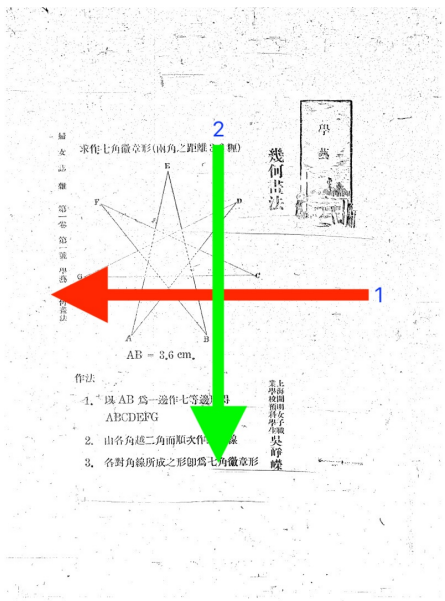
- For L1, the text reading rule is clear.
- The algorithm orders the text lines in the upper part and lower part separately, then combines them.



Text Ordering

- For L2, the text reading rules are diverse.
- But in general, vertical lines are read from right to left, and horizontal lines are read from top to bottom.
- The algorithm orders the vertical text lines and horizontal text lines separately, then combines them.





Text Ordering

- **Pros**: simple and effective. The major layout L1 can be ordered accurately, while the minor but diverse layout L2 is ordered by a general rule to prevent overfitting.
- **Cons**: need hyperparameters to distinguish L1 and L2, and the general rule may not be accurate for all images.

Synthetic Data Generation

- We generate two types of synthetic data to train the text detection and recognition models respectively.
- The synthetic data simulates images of Funü Zazhi, with diverse text layouts and varying image quality.
- The synthetic data is generated **continuously** during the training process.

Synthetic Data: Basic Units

- We select 6,000 most frequent Chinese characters[8] as the basic units in the synthetic data.
- We use 12 fonts that are commonly used in Funü Zazhi to render each character.
- Different characters of a same font form a **text line**.

Figure: 12 font styles of a specific traditional Chinese character

[8] https://lingua.mtsu.edu/chinese-computing/statistics/

Synthetic Data for Detection

- Simulate both L1 and L2 layouts.
- For L1, divide the image into two parts, and place **vertical** text lines in each part.
- Random noises and **non-text elements**[9] are added to the images to simulate various backgrounds.

Figure: Synthetic images for detection with layout L1

[9] From Icons-50 dataset: `https://arxiv.org/abs/1807.01697`

Synthetic Data for Detection

- For L2, randomly place both **vertical** and **horizontal** text lines in the image.
- Each **text line** uses a random typeface and font size. In L1, the font size and typeface are fixed for each **image**.

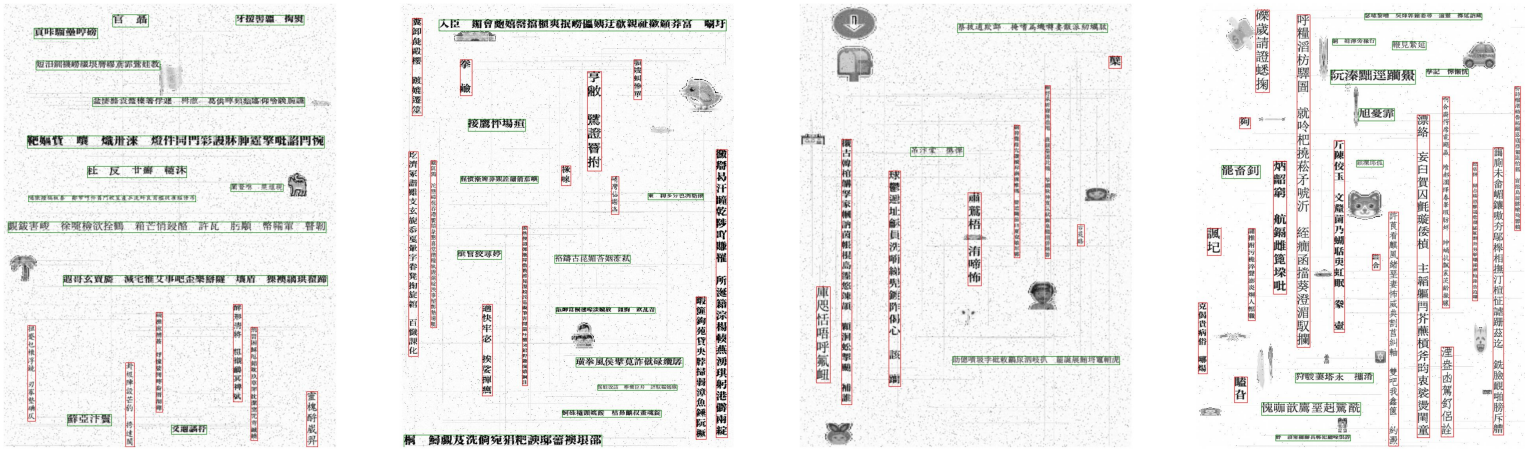


Figure: Synthetic images for detection with layout L2

Synthetic Data for Recognition

- Simulate text lines with varying lengths, font sizes, typefaces, and background noises.
- Frequency-based character selection[10].
- Replace characters with common words or phrases[11].

[10] https://lingua.mtsu.edu/chinese-computing/statistics/

[11] https://github.com/Embedding/Chinese-Word-Vectors

User Interface (Demonstration)

Test Dataset

- Manually selected 20 images from Funü Zazhi.
- Match the distribution of the whole dataset.
- Annotated with text line positions/labels and ordered text content.

Evaluation Metrics

- **Processing Time**: the inference time on test images.
- **mAP@[0.5:0.95]**: mean average precision at intersection over union thresholds from 0.5 to 0.95.
- **Character Error Rate** (CER): the edit distance between the predicted text and the ground truth.
- **BLEU-4** score: the average precision of 1-gram to 4-gram sequences.

Evaluation: Synthetic Data Generation

Technique to Remove	mAP@[0.5:0.95] (%)	Difference (%)
Baseline	69.9	-
Varying Character Size	67.0	**2.9**
Intensity Shift	68.4	1.5
Addictive Noise	68.8	1.1
Salt and Pepper Noise	68.7	1.2
Line-shaped Noise	68.9	1.0
Erasing Parts	69.5	0.4
Gaussian Noise	69.2	0.7
Non-text Elements	68.1	1.8

Table: Ablation study of synthetic data generation for detection

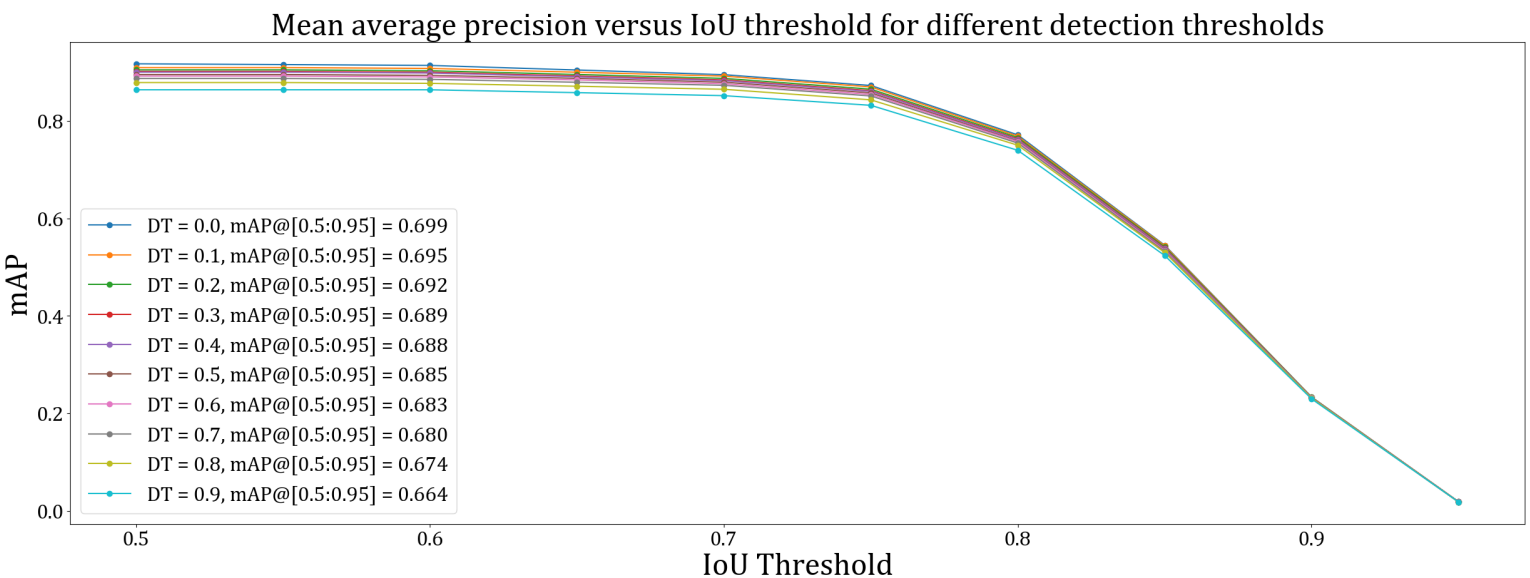
Evaluation: Synthetic Data Generation

Technique to Remove	CER	BLEU-4	Difference (%)
Baseline	0.328	0.546	-
Frequency-based Character Selection	0.401	0.479	**7.3** / 6.7
Word Replacement	0.389	0.462	6.1 / **8.4**
Varying Character Size	0.352	0.517	2.4 / 2.9
Intensity Shift	0.345	0.528	1.7 / 1.8
Addictive Noise	0.334	0.539	0.6 / 0.7
Salt and Pepper Noise	0.339	0.534	1.1 / 1.2
Line-shaped Noise	0.342	0.531	1.4 / 1.5
Erasing Parts	0.331	0.541	0.3 / 0.5
Gaussian Noise	0.332	0.540	0.4 / 0.6

Table: Ablation study of synthetic data generation for recognition

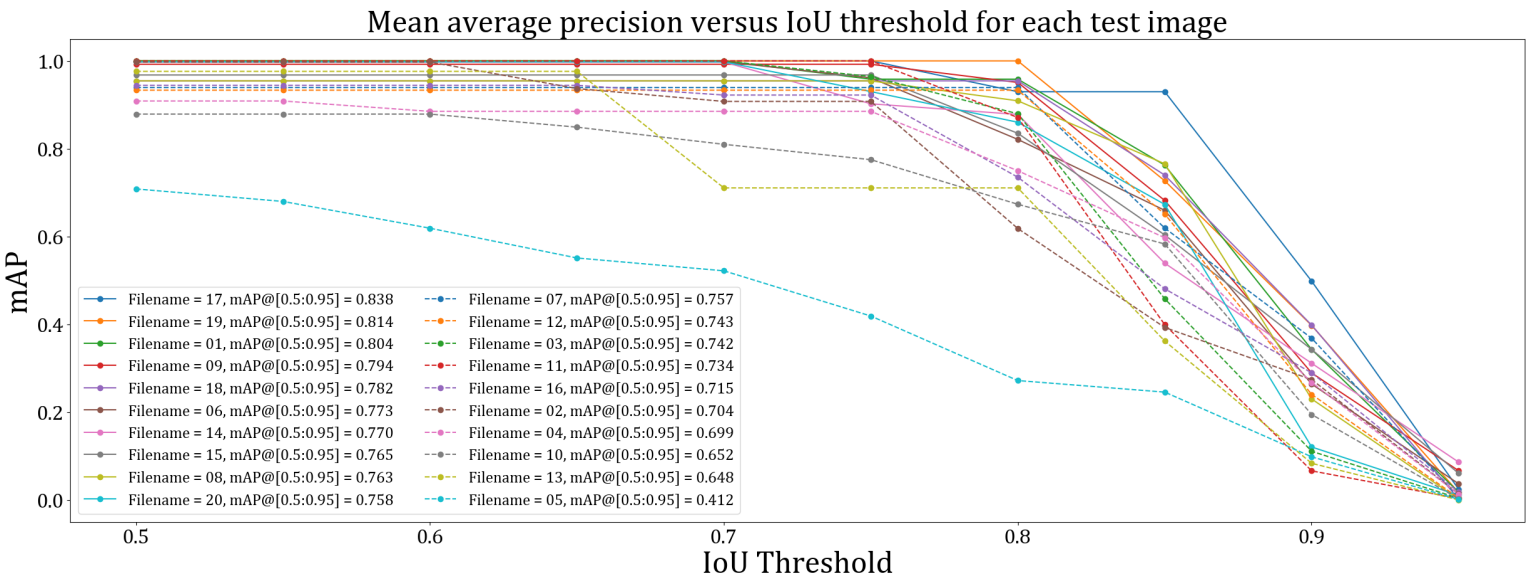
Evaluation: Text Detection

- Detection threshold filters out low-confidence text lines.
- A lower detection threshold, a higher mAP.
- mAP remains high for large IoU thresholds (0.8).



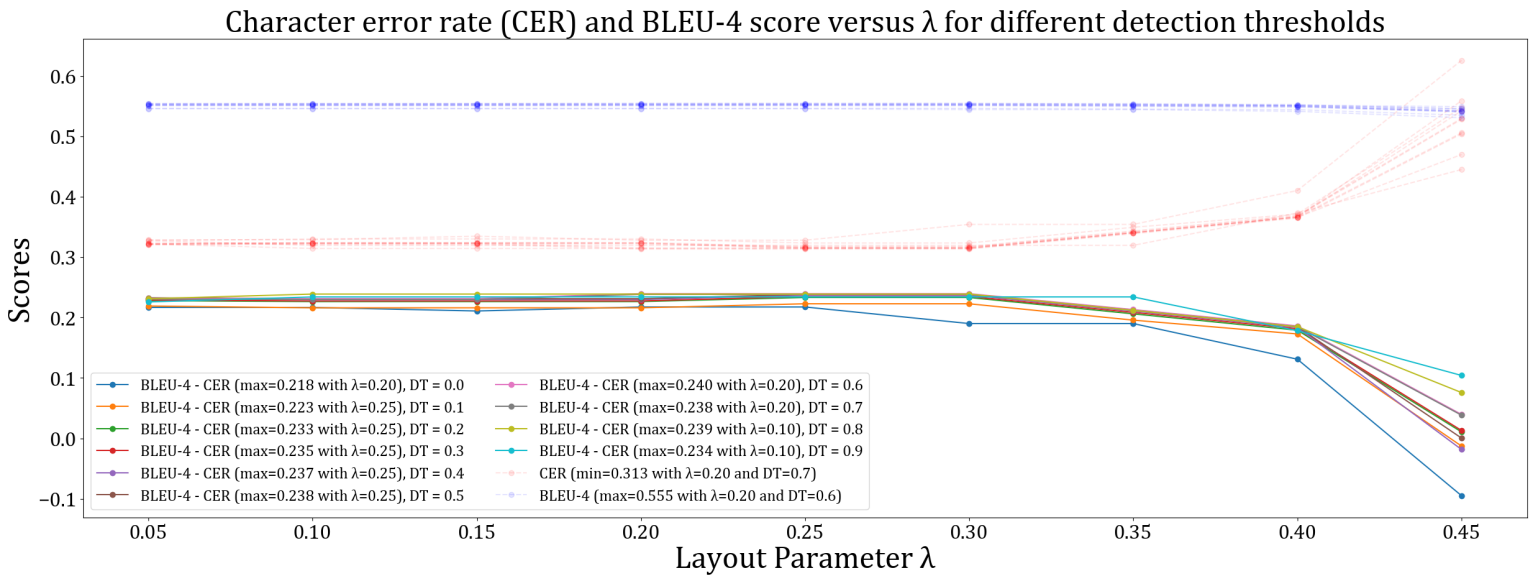
Evaluation: Text Detection

- Most images have mAP@[0.5:0.95] > 0.7.
- Only one image has mAP@[0.5:0.95] < 0.5.



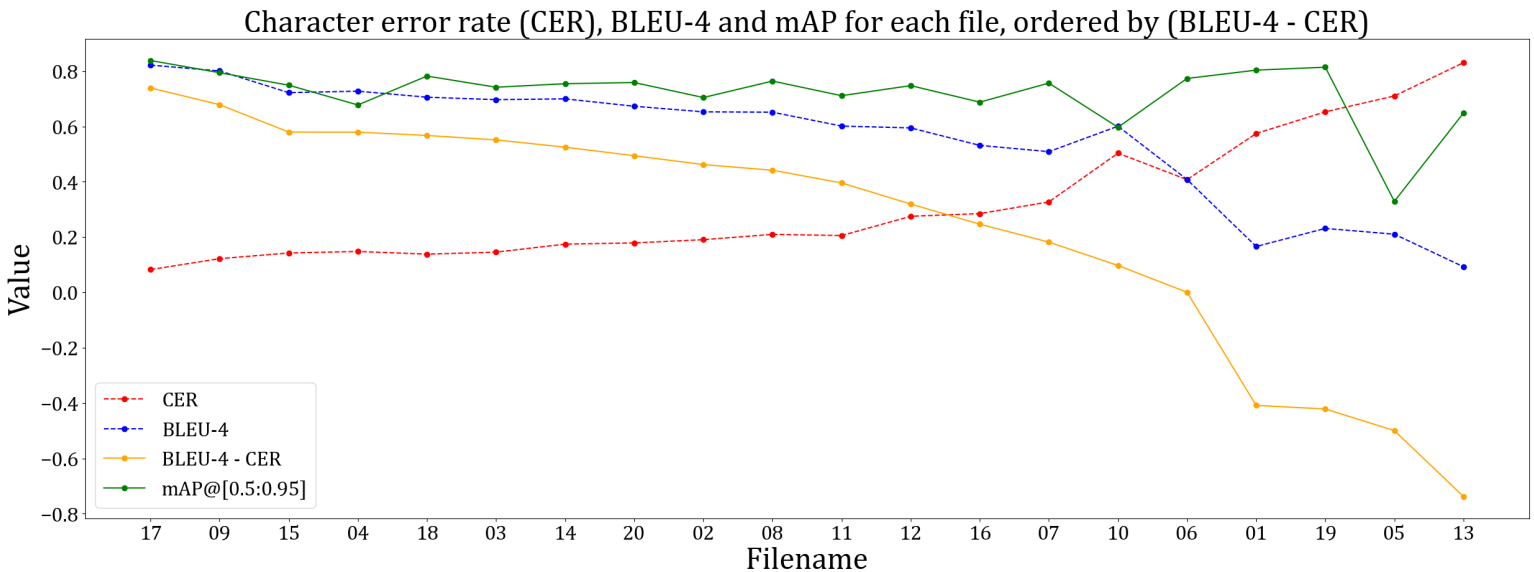
Evaluation: Text Recognition

- A higher λ indicates a stricter threshold to classify the text layout as L1.
- $\lambda = 0.2$, DT $= 0.6$ or 0.7 gives the best results.



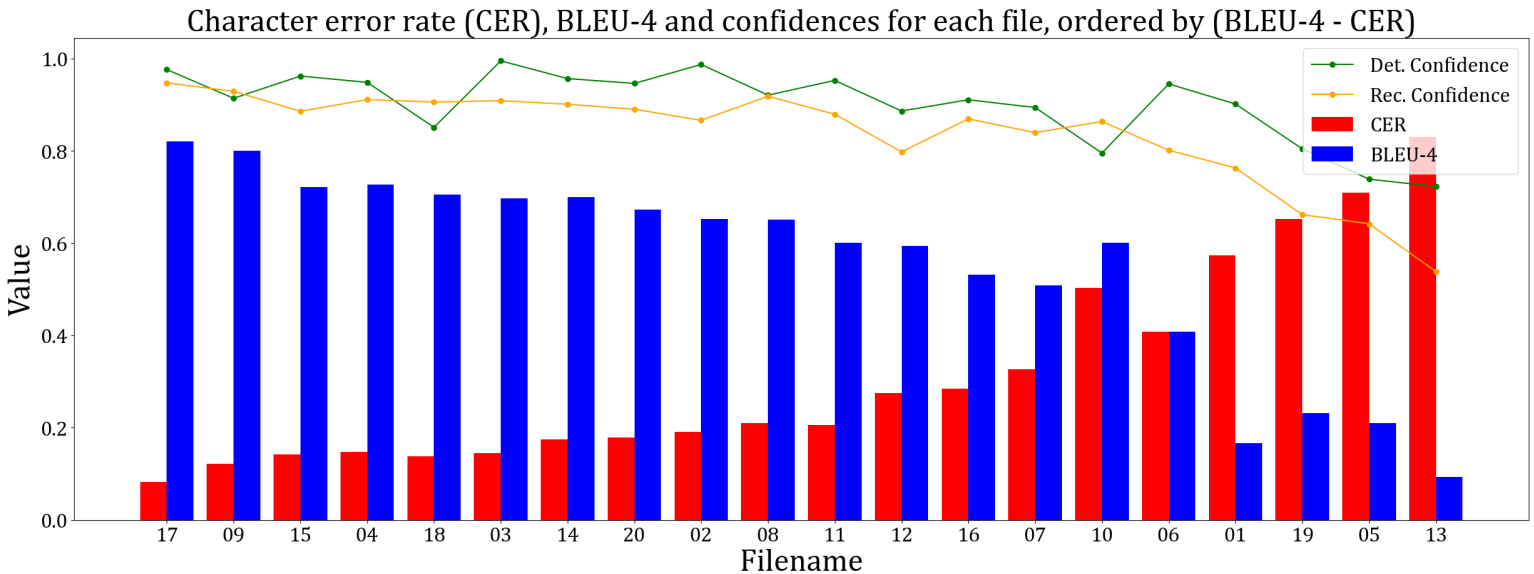
Relation between Metrics

- CER and BLEU-4 scores are negatively correlated.
- mAP is weakly correlated with CER and BLEU-4.
- Recognition performance is not directly related to detection performance.



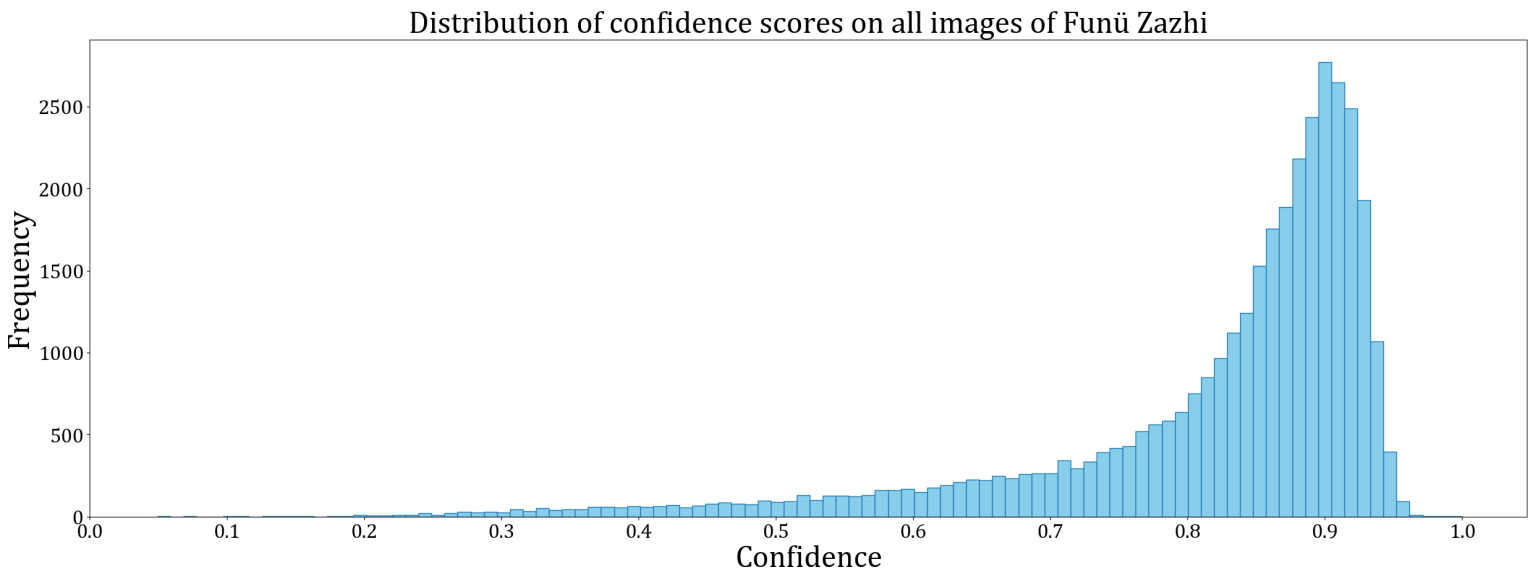
Confidence Scores

- **Estimate** the performance without ground truth.
- The output **probability** of a prediction.
- **Recognition confidence** is more related to overall performance than detection confidence.



Confidence Scores

- About 30% of the results with confidence above 0.9, 70% above 0.8, similar to the distribution in test set.
- Results with confidence above 0.9 in the test set achieve average CER of **0.15** and BLEU-4 of **0.73**.
- Around 10,000 images can achieve similar performance!



State of the Art

- Pre-trained and support traditional Chinese recognition.
- PaddleOCR[12]: latest release in 08/2024.
- EasyOCR[13]: latest release in 09/2023.
- Tesseract[14]: latest release in 12/2021.
- Apple MacOS Preview[15]: built-in OCR tool in MacOS.

[12] https://github.com/PaddlePaddle/PaddleOCR

[13] https://github.com/JaidedAI/EasyOCR

[14] https://github.com/tesseract-ocr/tesseract

[15] https://support.apple.com/en-gb/guide/preview/welcome/mac

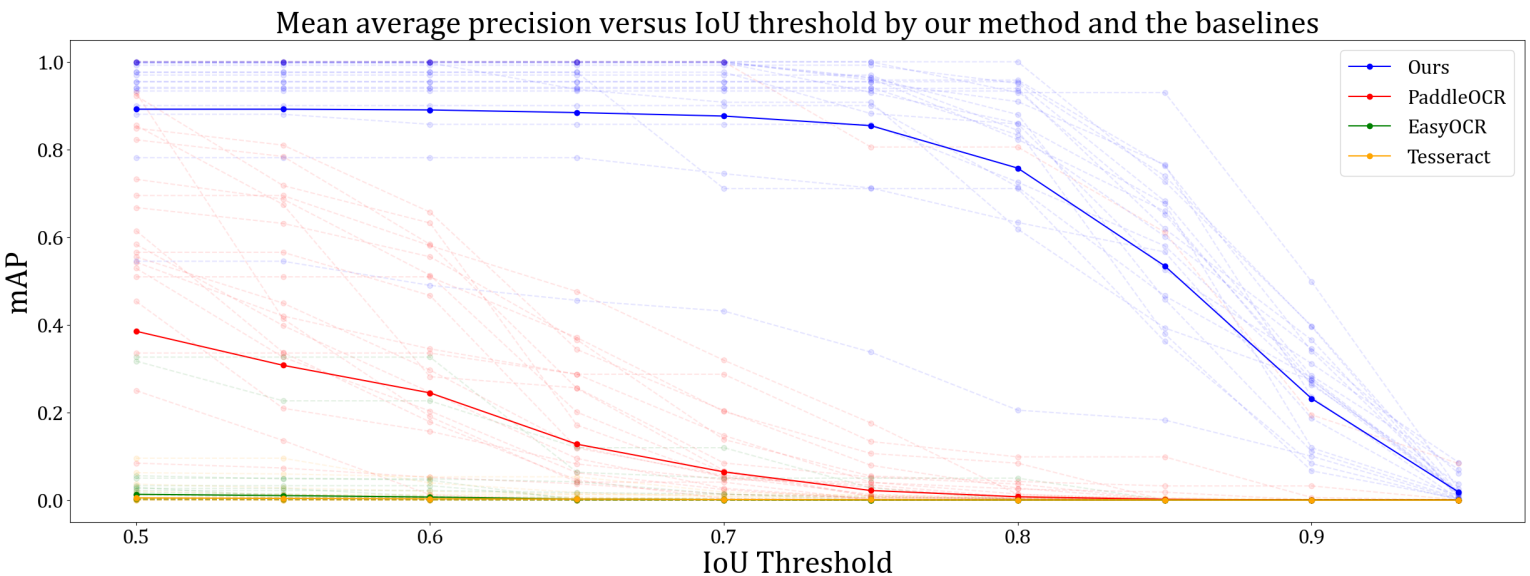
Comparison: Processing Time

- CPU: Apple M1 with 8GB RAM.
- GPU: NVIDIA GeForce GTX 1650 with 4GB VRAM.

OCR Method	Total Time (s)	Average Time per File (s)
PaddleOCR	111.27	5.56
EasyOCR	104.19	5.21
Tesseract	71.63	3.58
Ours	84.85	4.24
Ours (GPU)	**57.53**	**2.88**

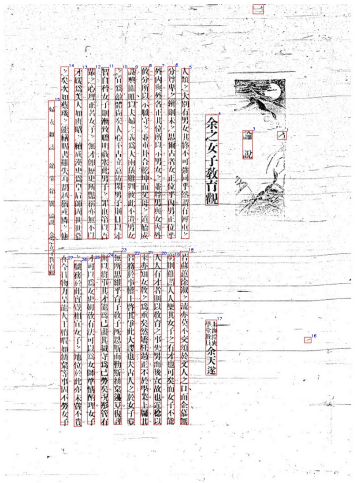
Comparison: Text Detection

- Outperforms all SOTA tools in mAP@[0.5:0.95].
- Apple MacOS Preview (AMP) not included as it cannot output text line positions.

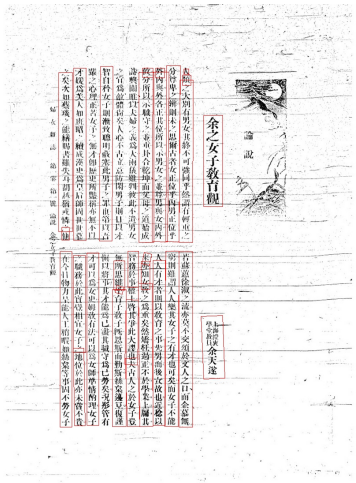


Comparison: Text Detection

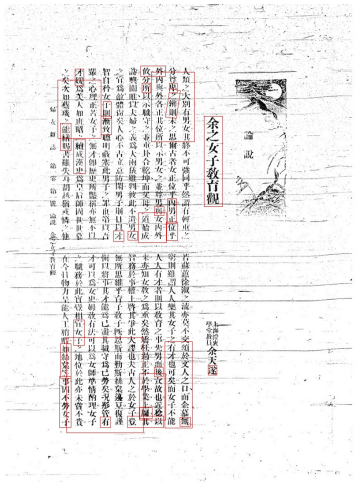
- Noticeably better detection performance.
- It represents the average image quality of Funü Zazhi.



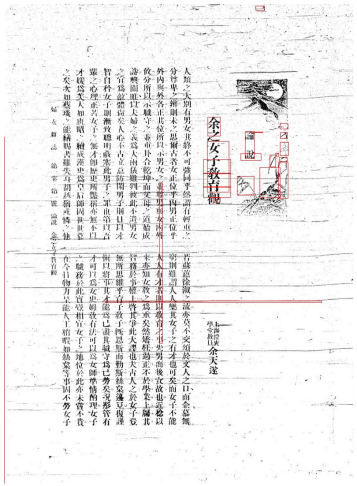
(a) Ours



(b) PaddleOCR



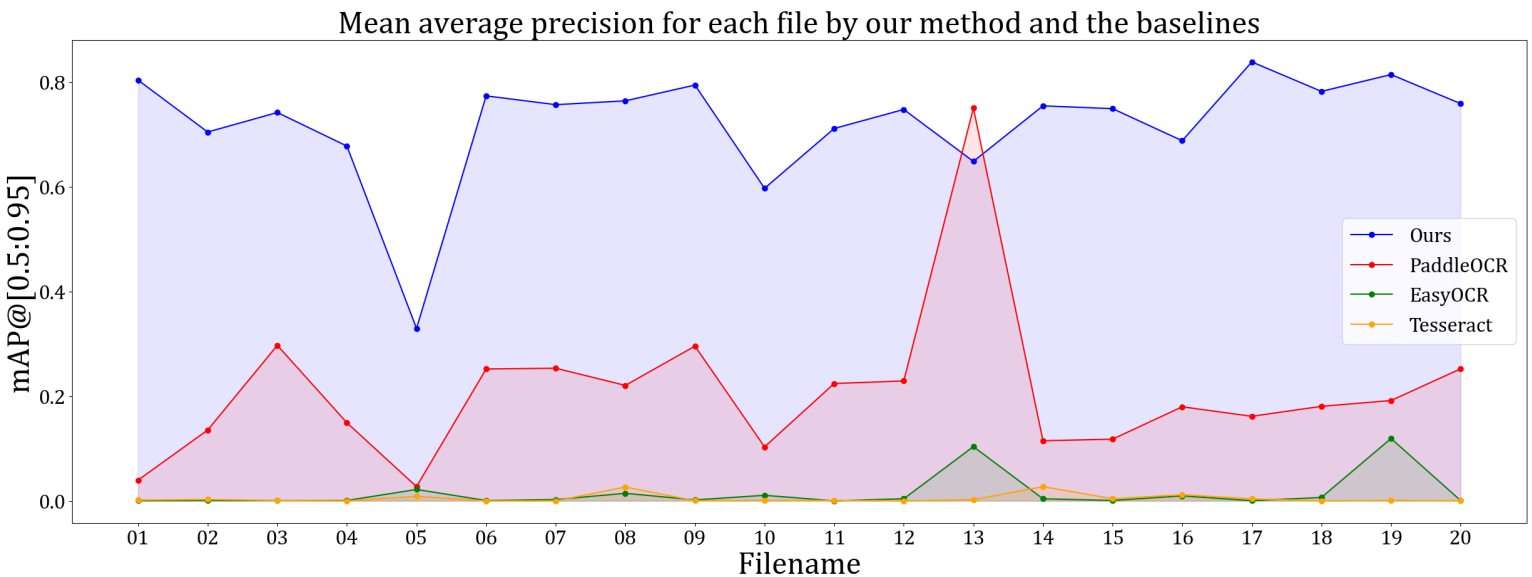
(c) EasyOCR



(d) Tesseract

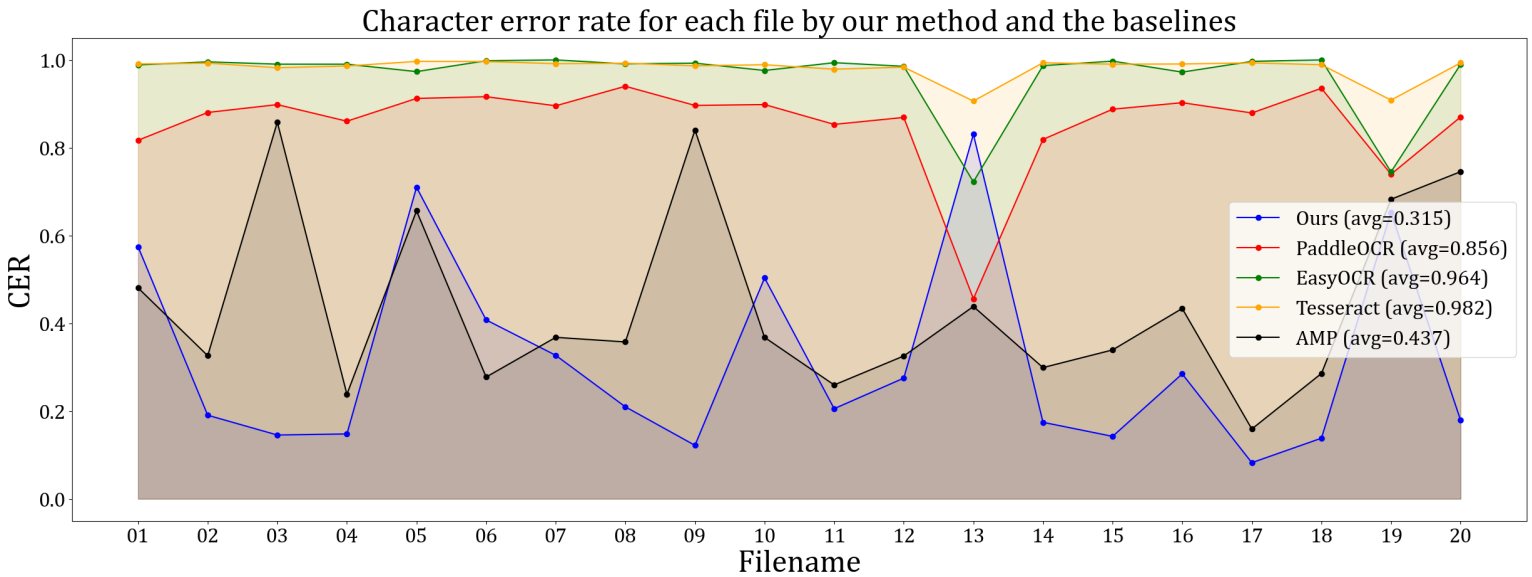
Comparison: Text Detection

- Results are more stable across different images.
- Only one image worse than PaddleOCR.



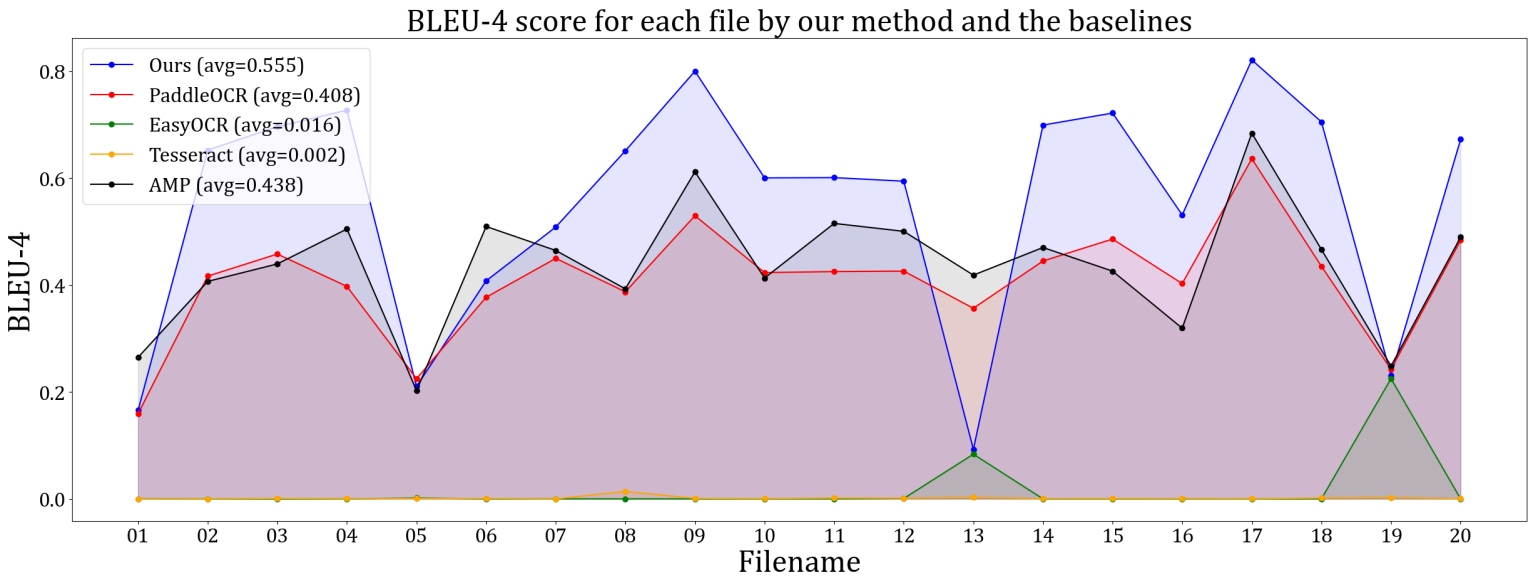
Comparison: Text Recognition

- The average CER is 12% lower than AMP.



Comparison: Text Recognition

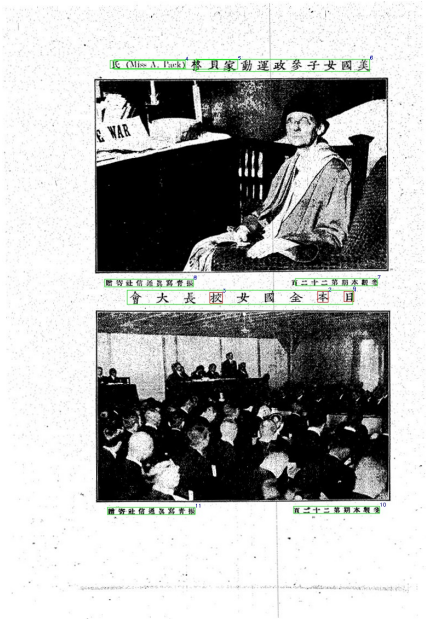
- The average BLEU-4 is 12% higher than AMP.



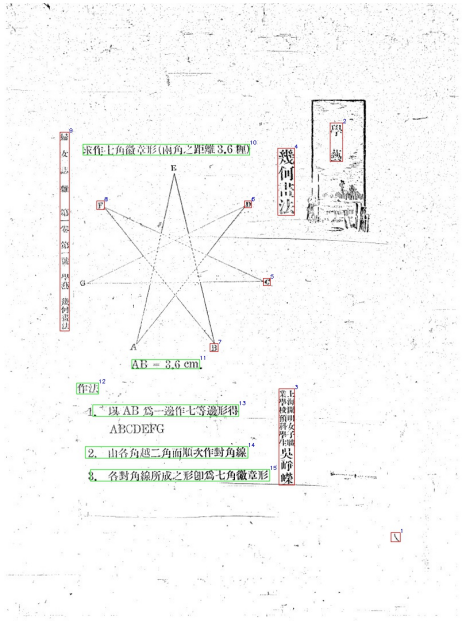
Limitations

- Complex text layouts (a), large character spacing (b), horizontal text lines (b), and non-Chinese characters (c).

(a) 05



(b) 13



(c) 19

Conclusion

- Developed an OCR system optimized for Funü Zazhi.
- Designed synthetic data to simulate Funü Zazhi images for training text detection and recognition models.
- Converted 36,101 images into text data, with about 30% achieving CER of 0.15 and BLEU-4 of 0.73.
- Achieved superior performance compared to 4 SOTA OCR tools on the test dataset.

Future Work

- Collect real annotated data for fine-tuning.
- Improve text ordering algorithms.
- Enhance synthetic data generation.
- Evaluate on relevant datasets.
- Extend to other historical documents.
- Deploy the system for public use.

Q&A

Thank you!