Detection and Recognition of Sino-Nom characters on woodblock-printed images

Trong Tuan Dao
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi, Vietnam
19020477@vnu.edu.vn

Cong Thuong Le
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi, Vietnam
thuonglc@vnu.edu.vn

Thi Duyen Ngo*
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi, Vietnam
duyennt@vnu.edu.vn

Thanh Ha Le
Faculty of Information Technology
VNU University of Engineering and Technology
Hanoi, Vietnam
ltha@vnu.edu.vn

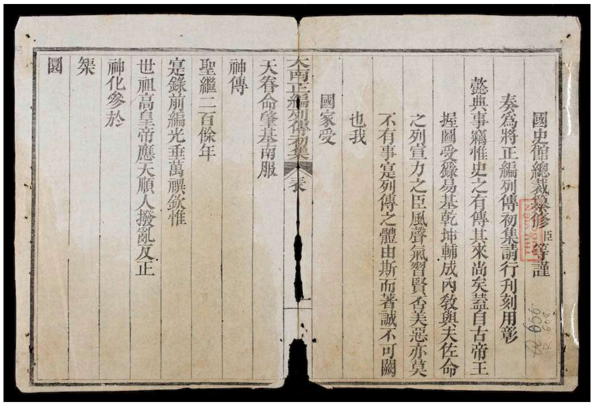
***Abstract*—There are many documents written in Sino-Nom capturing a wide range of historical, political, and literary facets of Vietnamese culture. To contribute to the process of preserving those documents, we focus on the task of building Optical Character Recognition models for Sino-Nom characters using a semi-supervised approach. Also, we propose a pipeline for collecting needed Sino-Nom document images which are used for training and evaluating our models. Being evaluated on our collected dataset, our OCR baseline achieves F1 scores of over 0.97 in detection and top-1 and top-5 accuracies of 80.1% and 90.1% in recognition, respectively.**

***Index Terms*—Sino-Nom OCR, Sino-Nom data collecting, Sino-Nom text detection, Sino-Nom text recognition**

I. Introduction

Chữ Nôm (Sino-Nom), an ancient ideographic vernacular script, was adopted as the national script of Vietnam after its independence from China in 939 CE. Throughout the next millennium from the 10th century to the 20th, Sino-Nom became the primary script for Vietnamese literature, philosophy, history, law, medicine, religion, and government policy. Sino-Nom woodblock is a type of document record created by carving Sino-Nom document content in reverse onto a board of wood. These woodblocks were used to print books in Vietnamese in the 19th and early 20th centuries, as depicted in Figure 1. The contents of these woodblocks vary from official literature to classic and historical books.

Digitization provides a way to preserve these Sino-Nom documents for future generations and to make them accessible to scholars and students around the world. One major problem with the digitization process is that it is a very time-consuming task and there is a shortage of Sino-Nom specialists. Thus, Optical Character Recognition (OCR) systems can be applied to contribute to this digitization process. To make use of those OCR systems, there is also a need for a considerable amount of labeled data to effectively train machine learning models. In this paper, we are proposing a pipeline for collecting Sino-Nom document images. Using the pipeline, we manage to construct a dataset for our OCR tasks, which is to build a two-stage model for the detection and recognition of Sino-Nom characters in document images.



國史館總裁纂修臣等謹
奏爲將正編列傳初集請行刊刻用彰
懿典事竊惟史之有傳其來尚矣蓋自古帝王
握圖受籙易基乾坤輔成內教與天佐命
之列宣力之臣風聲氣習賢否美惡亦莫
不有事寔列傳之體由斯而著誠不可闕
也我
國家受
大南正編列傳初集 表
天眷命肇基南服
神傳
聖繼二百餘年
寔錄前編光垂萬禩欽惟
世祖高皇帝應天順人撥亂反正
神化參於
架
圜

Figure 1. Example of a Sino-Nom page

II. Related works

To address the problem of detecting and recognizing Sino-Nom characters, there have been many experiments conducted in the past years. Tokyo University of Agriculture and Technology proposed many studies regarding Nom script recognition. ([13], [7]) In [13], the Nom characters are segmented by the X-Y cut method and the Voronoi diagram method, and then the segmented patterns are recognized using generalized learning vector quantization (GLVQ) and modified quadratic discriminant function (MQDF). To further improve the recognition rate, in [7], the authors present another approach using a deep CNN model combining the feature

* Corresponding author

extractor, the coarse category classifier based on the VGG network, and the fine category classifier including the latest inception layers. Using these methods, the top-1 recognition rate has improved significantly (from 69.08% to 81.73%). Those results are computed upon 32,695 classes of Nom characters. More recently, Vu et al. [14] have introduced the IHR-NomDB dataset of hand-written Sino-Nom characters. This dataset consists of over 260 pages that are collected from the Vietnamese Nom Preservation Foundation and manually analyzed and labeled. They also created a Synthetic Nom String dataset consisting of 101,621 automatically generated images using a collection of Nom sentences. The authors used a Convolution Recurrent Neural Network (CRNN) model with pre-training on the Synthetic Nom String dataset to test on a validation set of the handwritten dataset, achieving an accuracy of 42.70% at the string level and 82.28% at the character level.

Given the fact that the Sino-Nom script is a part of the Sinoxenic system, which is derived from the usage of the Chinese language by ancient Vietnamese, it is essential to consult studies on Chinese character recognition as well. In traditional supervised transfer learning, Li et al. [5] adopted a linear style transfer mapping method in the task of historical Chinese character recognition. In their study, Zhang et al. [16] incorporated an unsupervised adaptation layer into their network to effectively adjust the variability of writing styles in tasks involving the recognition of handwritten Chinese characters. In the semi-supervised transfer learning methods whose training data consists mostly of unlabeled samples and a few labeled samples, Tang et al. [12] adapted the Multi-Kernel Maximum Mean Discrepancy (MK-MMD) loss into the traditional CNN models. In their work, the recognition CNN model after being fine-tuned on the target domain is then trained by both a large number of unlabeled samples and the limited labeled samples in the target domain to minimize the MK-MMD loss. This work shows the potential usage of those semi-supervised approaches, hence we decided to adopt a semi-supervised approach into our pipeline, as will be described in later chapters.

III. Dataset

Overall, the collection of images of the Sino-Nom woodblock prints dataset was obtained by merging information from two online sources, namely Nomfoundation [1] and Wikisource [15]. The pipeline for collecting and labeling Nôm document pages is described in Figure 2

A. Data collection

We consider two websites as our sources of data. The first one, namely NomFoundation [1], is the official page of the Vietnamese Nôm Preservation Foundation. The second one, namely Wikisource [15], is a volunteer-run online library of source texts, which is accessible to all for free. NomFoundation website has a large collection of Sino-Nom document images, but most of them do not come in handy with their respective labels. Meanwhile, Wikisource offers a wide range of source texts for historical Nôm documents.

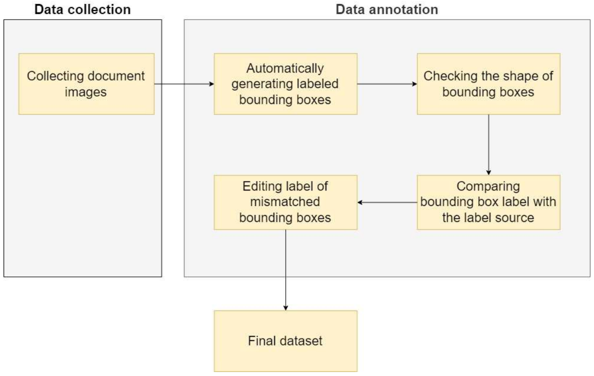

Data collection
Data annotation
Collecting document images
Automatically generating labeled bounding boxes
Checking the shape of bounding boxes
Comparing bounding box label with the label source
Editing label of mismatched bounding boxes
Final dataset

Figure 2. Dataset building pipeline

1) Collecting document images from NomFoundation: VNPF's official website offers a massive collection of scanned images of Sino-Nom document books. Those books cover a wide range of genres, from classics, history to philosophy and literature. To collect those precious data sources, BeautifulSoup [10] is leveraged to examine the VNPF Web API and to extract document image URLs for downloading.

2) Collecting document labels from Wikisource: After examining the Nôm historical documents available on Wikisource, we found several documents whose source texts are accurate and sufficient, such as *Dai Nam chinh bien liet truyen so tap*, *Dai Nam nhat thong chi* and *Kham dinh Viet su thong giam cuong muc*. We collected these documents as labels for the mentioned document images.

B. Data annotation

For the tasks of document annotation, we leverage the use of PaddleOCR and PPOCRLabel to annotate Nôm pages semi-automatically. PaddleOCR [8], developed by Baidu based on the PaddlePaddle deep learning framework, is an OCR framework that provides multilingual practical OCR tools, while PPOCRLabel [8] is a semi-automatic graphic annotation tool with built-in PaddleOCR model to automatically detect and re-recognize data.

After collecting the document images from the Nôm Foundation website, each image is then labeled using PPOCRLabel. To reduce the amount of time and effort needed in the labeling process, a semi-automatic process is adopted, in which a text detection and recognition model is utilized to automatically predict bounding boxes of characters with their respective labels in each document image, which acts as a draft for the annotator to manually edit later. Next, the annotator would check and edit the shape of the bounding boxes into a rectangle shape whose sides are parallel to the image sides. The annotator would then check these labels by manually comparing each character in the Nôm foundation document images with its equivalent Unicode character from the Wikisource page, as shown in Figure 3. The mismatched labels would be corrected by then.

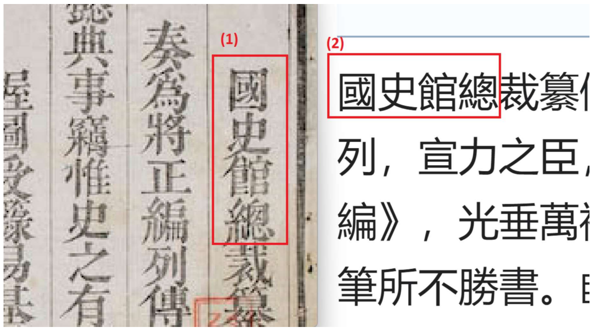


Figure 3. Example of equivalent characters from Nôm pages (1) and Wikisource sources (2)

C. Resulted dataset

At the end of the labeling process, we have gathered a dataset of 100 document pages with over 15000 Nom characters. These Nôm pages are from 3 books, i.e. *Dai Nam chinh bien liet truyen so tap*, *Dai Nam nhat thong chi* and *Kham dinh Viet su thong giam cuong muc*. Apart from those labeled data, we also gather Nôm pages with consistent printed quality to use in the semi-supervised learning model, which is from *Dai Nam chinh bien liet truyen so tap*, *Dai Nam nhat thong chi*, *Kham dinh Viet su thong giam cuong muc chinh bien*, *Dai Nam thuc luc tien bien*, *Dai Viet su ky toan thu* and *Quoc trieu chinh bien toat yeu*. Those collected pages can also be annotated by our proposed pipeline. Details of labeled and unlabeled dataset is noted in Table I and Table II

Table I
Statistics of labeled and unlabeled pages

Book Title	**Number of labeled pages**	**Number of unlabeled pages**
Đai Nam chinh bien liet tuỷen so tap	63	435
Dai Nam nhat thong chi	20	112
Kham dinh Viet su thong giam cuong muc	17	382
Dai Nam thuc luc tien bien	-	348
Dai Viet su ky toan thu	-	569
Quoc trieu chinh bien toat yeu	-	419
Total	**100**	**2265**

Table II
Statistics of labeled and unlabeled characters data.

Status	**Number of Pages**	**Number of Characters**	**Percentage of Characters (%)**
Labeled	100	15986	2.50
Unlabeled	2265	624226	97.50
Total	2365	640221	100.00

In the dataset, 677 different characters appear only once, while the most frequent character appears 370 times. More information about frequencies of characters in the dataset is shown in Table III

Table III
Character frequencies in the labeled dataset.

Frequency intervals	**Number of Different characters**
1	667
2-5	647
6-10	221
11-50	304
51-100	36
>100	15
Total vocabularies	**1890**

IV. Methods

Overall, our OCR pipeline consists of two stages, as shown in Figure 4. For the former stage, YOLOv5 is used as our detection model which accepts the Nôm pages as its input and outputs separate cropped images of Nôm characters. Then for the latter stage, we adopt a semi-supervised transfer learning approach through the help of the SimCLR Framework. We use mostly unlabeled samples with a few labeled samples to train a classification model that classifies each Nôm character image into its corresponding label.

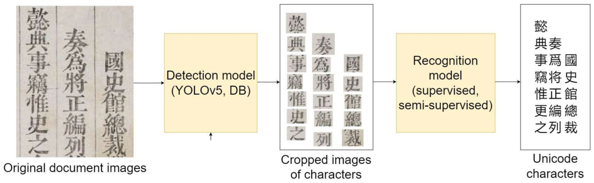

Original document images
Detection model (YOLOv5, DB)
Cropped images of characters
Recognition model (supervised, semi-supervised)
Unicode characters

Figure 4. Overview OCR pipeline

A. Bounding box detection

For the Nôm character detection task, we try and conduct experiments to compare two major approaches to the general object detection problem, i.e. the regression-based approach and the segmentation-based approach. While regression-based approaches aim to train a regression model to directly achieve the locations of objects [17], segmentation-based approaches tend to calculate a probability map recording the probability of each pixel being the object needed for detection. We compare two methods as representatives of the two above approaches, which are YOLO and DBNet.

1) YOLO: YOLO [9] (You Only Look Once) is a state-of-the-art object detection algorithm that was introduced in 2016 by Joseph Redmon et al. The YOLOv5 (YOLO version 5) [4] architecture is a fully convolutional neural network that uses a backbone feature extractor and a set of detection heads to predict object bounding boxes and class probabilities. YOLOv5 models are implemented on the widely-used platform PyTorch, making them easier to use.

2) DBNet: DB (Differentiable Binarization) is a module proposed by Liao et al. [6] aiming to insert the binarization operation into a segmentation network for joint optimization. As the DB module is fully differential, the process of binarization is trainable and can be included in an end-to-end convolutional neural network. In the DB module, the input image is first fed into a feature-pyramid backbone. The pyramid features at different scales are then upscaled and concated to form the feature F. The feature F is used to predict both the probability map and the threshold map. In the following, the binarization map is approximated by the probability map and the threshold map. The module architecture is shown in Figure 5

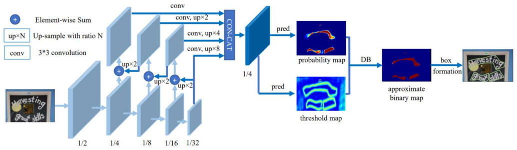


Figure 5. Architecture of DB Module from original paper [6]

B. Recognition using semi-supervised classification model

For the recognition task, we consider it a classification problem with each Sino-Nom character as a label. A semi-supervised approach using SimCLR is examined in which we rely on both annotated and unannotated data to train the recognition model effectively. The system architecture for a semi-supervised approach to the recognition task consists of two main steps, as can be seen in Figure 6. The former step is to train a contrastive learning model using only unlabeled input images. The output of this step is an encoder whose weighted parameters are used for a supervised classification model in the latter part. The latter part aims to fine-tune this classification model on the labeled data to provide the final OCR model. SimCLR [2] is utilized for the pre-training task. We follow the default setting proposed in the original paper to train a contrastive learning model, which learns visual representations of unlabeled input images. These representations are trained to minimize the NT-Xent losses of similar pairs and maximize those losses of dissimilar pairs.

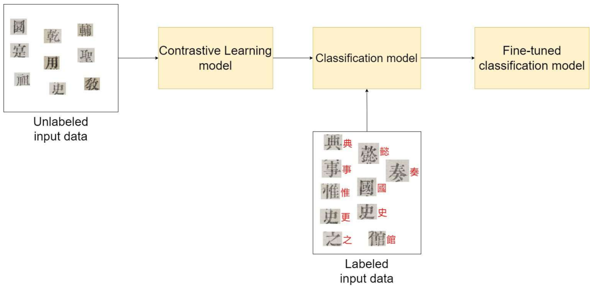


Figure 6. Semi-supervised Learning Model Architecture

The model architecture used for the pre-training task is mostly similar to the default setting proposed in the original paper of SimCLR, except for the data augmentation step. The default settings as proposed use various data augmentation techniques to create different views of input samples, including color jitter, random crop, random grayscale, Gaussian blur, random rotation, horizontal and vertical flip. We only apply 3 techniques from those mentioned above, i.e. the color jitter, random grayscale, and random rotation techniques.

SimCLR, as the authors proposed, can employ different neural network architectures as its base encoder, which learns an embedding space from augmented images. We conduct our experiment with ResNet [3] as the base encoder of our SimCLR implementation. In the fine-tuning task, our ResNet encoder is extracted from the contrastive learning model to be finetuned on the downstream task, i.e. the classification task of the Sino-Nom script, using the annotated data. We finetune only the last fully connected layer, which is responsible for mapping the learned features to class probabilities.

V. Experiments and Results

A. Environments

All experiments are run on the Google Colaboratory Pro platform. Some of its hardware specifications are listed as follows:

- CPU: Intel Xeon 2.3 GHz 8-core processor
- RAM: 27GB
- GPU: NVIDIA Tesla T4 (16 GB VRAM) or NVIDIA Tesla P100 (16 GB VRAM)
- Storage: 100GB of SSD storage

B. Detection experiments and results

For the detection task of character-level bounding boxes, different experiments are conducted using various pre-trained models of YOLOv5 and a pre-trained DB-based PaddleOCR model. All models are trained for 30 epochs over 90 out of 100 labeled document images, leaving 10 images for evaluation. The evaluated models of YOLOv5 are Nano model YOLOv5n, Small model YOLOv5s, Medium model YOLOv5m, Large model YOLOv5l, and Extra Large model YOLOv5x. For the YOLO models, the hyperparameters chosen are as follows:

- Batch size: 3
- Learning rate: 0.01
- Momentum: 0.937
- Weight decay: 0.0005
- Optimizer: SGD with Nesterov momentum
- Image size: 1280x1280
- IoU threshold: 0.2

For the DBNet models, the hyperparameters chosen are as follows:

- Batch size: 8
- Learning rate: 0.001
- Optimizer: Adam
- Box threshold: 0.6

The metrics used for model evaluation are Precision, Recall, and F1 score. Table IV shows the detailed results of those experiments.

As can be seen from the detection results, all detection models perform well within a few training epochs, all achieved F1 scores of over 0.97. Within the family of YOLO models, even though the extra large model YOLOv5x is considered

Table IV
Results of detection models

Model	Precision	Recall	F1 score	Epochs	Duration
YOLOv5n	0.9824	0.9693	0.9758	30	1h09m
YOLOv5s	0.9840	0.9702	0.9770	30	1h14m
YOLOv5m	0.9842	0.9712	0.9776	30	1h17m
YOLOv5l	0.9833	0.9700	0.9766	30	1h21m
YOLOv5x	0.9838	0.9721	0.9779	30	1h35m
DB	0.9864	0.9624	0.9743	30	1h24m

much larger and more powerful than other YOLOv5 models, our comparison shows that the speed of simpler models, like YOLOv5n or YOLOv5s, is a better trade-off than achieving a little more precision.

C. Recognition experiments and results

The unlabeled bounding boxes used for contrastive learning are inferred by applying the fine-tuned YOLOv5m model in Section V-B over the unlabeled document images, which results in 624,226 unlabeled bounding boxes. The contrastive learning SimCLR model is trained for 70 epochs before its weights are transferred into the fine-tuning step. For the contrastive learning model to learn different views of the same images, some data augmentation techniques are applied to the unlabeled dataset, including Random rotation by an angle within the range of (-15°; 15°), Random grayscale and Color jitter.

The labeled data of bounding boxes, consisting 15,995 cropped images of Sino-Nom characters, are divided into two sets, i.e. training set and testing set, by the ratio of 9:1. Since the labeled data used for the learning task is not sufficient, some augmentation techniques are applied onto the training set, such as Random rotation by an angle within the range of (-15°; 15°) and Grayscale, as shown in Figure 7. At the end of the data augmentation process, our training set consists of 43,161 images, and the test set consists of 1599 images. Both the neural network models in the supervised approach and the fine-tuned model in the semi-supervised approach are trained and evaluated on this set of labeled bounding boxes. The dataset labels are then encoded by using a vocabulary of 8733 characters combining the vocabulary of our dataset and a vocabulary of most frequent Chinese characters [11].

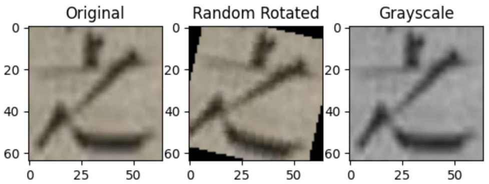


Figure 7. Example of data augmentation process

We use a fully-supervised model with ResNet architecture as our baseline to compare our semi-supervised model method. The results after training both models for 20 epochs, computed as top-1 and top-5 accuracy are shown in Table V.

Table V
Results of recognition models

Model	Top-1 Accuracy(%)	Top-5 Accuracy(%)
Supervised ResNet	71.1757	82.5391
Semi-supervised Model	80.0938	90.1032

As can be seen from the results, with the help of unlabeled data in the contrastive learning step, the Semi-supervised Model performs much better on the test set with a top-1 accuracy of 80.0938% and top-5 accuracy of 90.1032%. Among the false predictions in both models, we found that most of the characters that appear only a few times in the training set may end up being failed to predict by our models. The distributions of false prediction characters in the training set of the two models can be seen in Figure 8. Figure 9 visualized some false predictions made by our models of some characters that appear more frequently in the training set. The problem may be either the quality of printed characters or mistakes in the hand-cropped process.

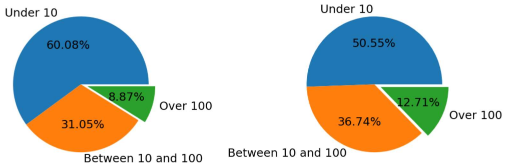


Figure 8. Distributions in the training set of false predictions characters made by our semi-supervised model (left chart) and supervised model (right chart)

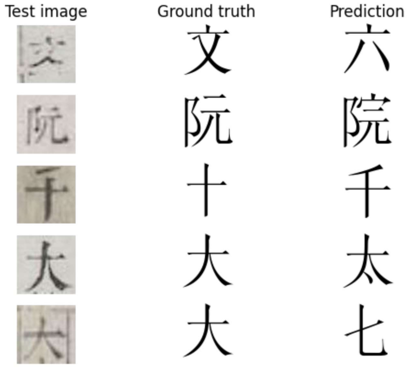


Figure 9. Example of false predictions

VI. Conclusion

In this paper, we present a pipeline to construct a Sino-Nom dataset by combining two different sources. Using this pipeline, we manage to build a labeled dataset of 100 Sino-Nom pages with 15,996 characters. Also, we propose an approach to deal with Sino-Nom character detection and recognition problems using YOLO and DB for Sino-Nom character detection and using SimCLR for semi-supervised learning to recognize Sino-Nom characters. Our detection models achieve F1 scores of over 0.97 and our recognition model achieves top-1 and top-5 accuracies of roughly 80.1% and 90.1%, respectively.

Given the promising results, our work could be extended in several ways. Using our proposed pipeline, a bigger Sino-Nom dataset can be constructed, which results in better models in the detection and recognition of characters in the mentioned language. More experiments should be carried out to better improve the performance of the models based on the potential data.

Acknowledgement

This work has been supported by Korea-Vietnam joint research project number NĐT/KR/21/17 - "AI-based Application for 3D-Model Restoration of Vietnamese and Korean Cultural Heritage".

References

[1] John Balaban, Lee Collins, Stephen Lesser, John Phan, D. Neil Schmid, and Ngô Trung Việt. *Vietnamese Nôm Preservation Foundation*. URL: http://nomfoundation.org/.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[4] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, (Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. Version v7.0. Nov. 2022. DOI: 10.5281/zenodo.7347926. URL: https://doi.org/10.5281/zenodo.7347926.

[5] Bohan Li, Liangrui Peng, and Jingning Ji. "Historical Chinese character recognition method based on style transfer mapping". In: *2014 11th IAPR International Workshop on Document Analysis Systems*. IEEE. 2014, pp. 96–100.

[6] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. "Real-time scene text detection with differentiable binarization". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 11474–11481.

[7] Cong Kha Nguyen, Cuong Tuan Nguyen, and Nakagawa Masaki. "Tens of thousands of nom character recognition by deep convolution neural networks". In: *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*. 2017, pp. 37–41.

[8] *PaddleOCR Github repository*. URL: https://github.com/PaddlePaddle/PaddleOCR.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[10] Leonard Richardson. *Beautiful soup documentation*. 2007.

[11] *Table of General Standard Chinese Characters*. URL: https://www.gov.cn/zwgk/2013-08/19/content_2469793.htm.

[12] Yejun Tang, Bing Wu, Liangrui Peng, and Changsong Liu. "Semi-supervised transfer learning for convolutional neural network based Chinese character recognition". In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. IEEE. 2017, pp. 441–447.

[13] Truyen Van Phan, Kha Cong Nguyen, and Masaki Nakagawa. "A Nom historical document recognition system for digital archiving". In: *International Journal on Document Analysis and Recognition (IJDAR)* 19 (2016), pp. 49–64.

[14] Manh Tu Vu, Van Linh Le, and Marie Beurton-Aimar. "Ihr-nomdb: The old degraded vietnamese handwritten script archive database". In: *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III 16*. Springer. 2021, pp. 85–99.

[15] *Wikisource website*. URL: https://zh.wikisource.org/wiki/Wikisource:%E9%A6%96%E9%A1%B5.

[16] Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark". In: *Pattern Recognition* 61 (2017), pp. 348–360.

[17] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.