INT3404E 20 - Image Processing: Final Report (Group 9)

Nguyen Quang Huy*
ID: 22028077
22028077@vnu.edu.vn

Nguyen Quang Huy
ID: 21020204
21020204@vnu.edu.vn

Kieu Minh Khue
ID: 22028067
22028067@vnu.edu.vn

Mai Ngoc Duy
ID: 22028255
22028255@vnu.edu.vn

ABSTRACT

The recognition of Sino-Nom characters in historical texts is of paramount importance for the preservation and understanding of cultural heritage, as these characters represent a unique fusion of Chinese and Vietnamese linguistic elements. Accurate recognition of these characters not only facilitates the transcription and digitization of ancient manuscripts but also enables scholars to conduct more effective historical and linguistic research. In this paper, we present our analysis of two different datasets, and based on this analysis, we explored several methodologies to address the challenges. Our experimental results demonstrate that our best-performing approach achieves an accuracy of approximately 80%.

1 INTRODUCTION

The Sino-Nom character, a unique and fascinating script, represents a captivating blend of Chinese and Vietnamese linguistic and cultural influences. This fascinating writing system, developed and flourished in Vietnam from the 10th century to 19th century, serves as a testament to the historical and cultural ties between these two nations. While rooted in Chinese character (Hanzi), Sino-Nom diverged significantly, evolving into a distinct system that reflected the nuances of Vietnamese pronunciation and grammar. This adaptation involved not only adopting existing Hanzi characters but also creating new ones, often combining existing characters or modifying their strokes to represent specific Vietnamese sounds. This rich heritage, however is now nearly lost. Today, less than 100 scholars world-wide can read Sino-Nom [1]. Indeed, digitalizing all historical Sino-Nom documents is significantly vital for the preservation of this cultural heritage.

There are several researches about constructing datasets and handling with the detection and recognition of Sino-Nom characters. In this paper, we focus only on Sino-Nom recognition task. Our contributions can be summarised as follows:

(1) We conduct the analysis on the given dataset and NomNaOCR dataset [4].
(2) From the data analysis, we have two main approaches: ResNet for supervised learning in the given dataset and SimCLR for semi-supervised learning in our constructed dataset, which is a combination of provided dataset and preprocessed NomNaOCR.
(3) We conduct some experiments and analysis with the results of our approaches and we suggest some methods to improve the model accuracy.

2 RELATED WORK

Text recognition has been an extensively researched area in the field of computer vision. The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), has made a great improvement in this task. CNNs can automatically learn hierarchical representations from raw pixel data, enabling end-to-end image recognition. Architectures like AlexNet [13], VGGNet [17], GoogLeNet [18], ResNet [9], EfficientNet [19],... have achieved remarkable performance gains in large-scale image recognition tasks, such as ImageNet classification. Transfer learning, where pre-trained CNN models are fine-tuned on limited data, has further improved recognition accuracy and efficiency [10]. Semi-supervised and unsupervised learning methods, such as contrastive learning, have significantly advanced the handling of large unlabeled datasets [3]. These methods reduce the dependency on small labeled datasets during the training phase, allowing for rapid improvement in model performance by incorporating additional unlabeled data. Some approaches have integrated Recurrent Neural Networks (LSTM, Transformer,...) with CNNs to deal better with patches of text, speeding up the ability of processing sentences and paragraphs appearing in an image [4].

Deep learning have been extensively utilized in various character recognition tasks across different languages and domains. Some notable examples include Telugu handwritten character recognition [11], Kannada character recognition [12], Devanagari character recognition for Vanakri [14], Bangla character recognition [6], Indic handwritten character recognition [7], character reading from identity documents in India [16], Tamil character recognition from original palm leaf manuscripts [8], and the recognition of Chinese and Korean characters [2]. Several research studies have explored Sino-Nom character recognition, employing various techniques such as modified quadratic discriminant function [21], deep CNN [15, 20], CNN with Transformer [4], or integration of contrastive learning [5].

3 TASK DEFINITION OF SINO-NOM RECOGNITION

Given an image of a character extracted from a Sino-Nom document as input, the task is to classify the image into one of several classes representing several Sino-Nom characters in the provided dataset.

4 ANALYZING DATA

4.1 Provided dataset

The given training dataset has a total of 56,130 images, classified into 2,130 different labels (an average of 26.35 images/label), each

*Group leader
[1] https://www.nomfoundation.org/nom-script/What-is-Nom-

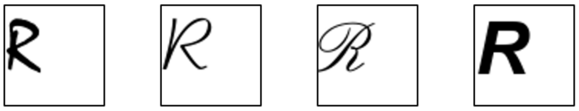
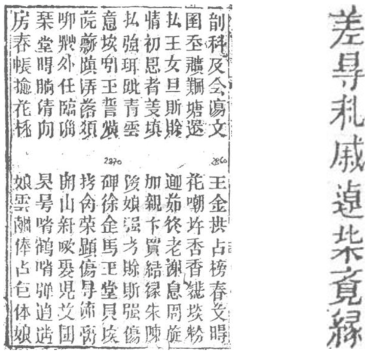


Figure 1: Noisy Data

comprising several images containing the same character but with different strokes. Moreover, many labels include low-quality images, which require preprocessing before applying them to the model. The majority of the labels have fewer than 50 images, and only a small number of them have more than 100 images. This uneven distribution of images across labels may lead to a decrease in the model's accuracy. We discovered that 1,660 labels include low-quality images, while the rest are composed of high-quality images. Furthermore, we discovered that images with label 1986 is label of letter R, not a Sino-Nom character (see Figure 1).

4.2 NomNaOCR

The NomNaOCR dataset [4] stands out as a comprehensive collection of Sino-Nom data, encompassing nine distinct documents comprising a combination of poetry and prose works. This dataset is divided into two primary types: pages, consisting of images capturing entire document pages, and patches, which are sequences of aligned characters extracted from those pages. All the pages and patches are labeled by the characters appearing in the images. The dataset statistics is illustrated in Table 1.



(a) (b)

Figure 2: Example of images in NomNaOCR. (a) Example of a page. (b) Example of a patch.

5 APPROACHES

5.1 Create custom datasets

From two datasets we analyzed, we decided to create custom datasets to match the data format of our task: SuperNom for supervised learning and SemiSuperNom for semi-supervised learning.

Documents	Pages	Patches
Luc Van Tien	104	2053
Tale of Kieu 1866	100	2385
Tale of Kieu 1871	136	3248
Tale of Kieu 1872	163	3254
DVSKTT-1 Quyen thu	105	929
DVSKTT-2 Ngoai ky toan thu	178	2605
DVSKTT-3 Ban ky toan thu	932	11118
DVSKTT-4 Ban ky thuc luc	787	7891
DVSKTT-5 Ban ky tuc bien	448	4835

Table 1: Statistics of NomNaOCR datasets

5.1.1 SemiSuperNom. We use previous Nom character localization tool [2] to automatically extract character images from pages in NomNaOCR. Then, we combine the images received after extraction with the provided dataset on the class to get final dataset for models following semi-supervised learning approaches.

In total, there are 408,792 images in SemiSuperNom datasets. However, this dataset is not high-quality because it contains a lot of noise images due to tool misrecognition. These noise images are mostly non-character images or characters which do not have an exact label in the original provided dataset.

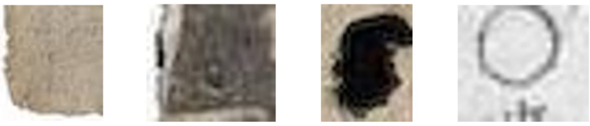


Figure 3: Example of noisy images in SemiSuperNom

5.1.2 SuperNom. To enhance the quality of the constructed dataset and utilize the available character labels in patches from NomNaOCR, we designed a semi-automatic labeling pipeline for SuperNom. Initially, for each numbered label in the class-provided dataset, we randomly selected a representative image. We then employed a character recognition tool from GitHub [3] to determine the character label corresponding to each numbered label. A team member manually verified these labels to ensure 100% accuracy in the number-to-character label mapping. This process resulted in a table mapping numbered labels to character labels. Assuming that the characters in a patch are equally spaced, we proceeded to vertically divide each patch by the number of characters indicated in the patch labels, mapping each resulting character image to its corresponding character label (we attempted to extract character with the same methods as we did in SemiSuperNom, however, due to the default localization method in the given tool, images received were all tiny and thus cannot use). Finally, we converted the character labels back to numbered labels, excluding any images with non-existent numbered labels. Combined with the class-provided dataset, we get a higher-quality SuperNom with all labeled images. An illustration of the pipeline is shown in Figure 4.

[2] https://github.com/trhgquan/OCR_chu_nom

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

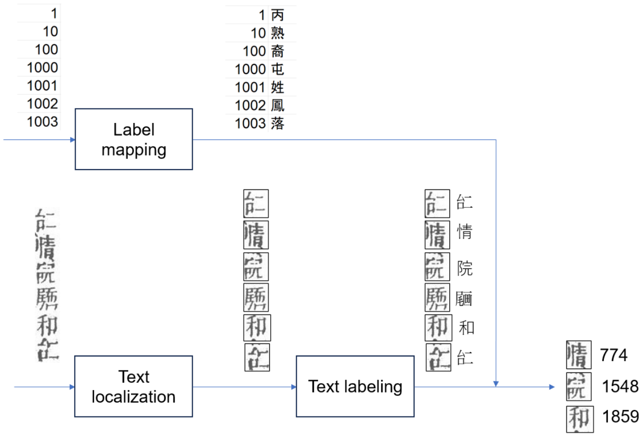


Figure 4: SuperNom dataset construction

In total, there are 387,516 images in SuperNom dataset. The images are in higher quality, but some images are still noisy due to worse character localization compared to the extraction method of SemiSuperNom.

Figure 5: Example of noisy images in SuperNom

5.2 ResNet

We choose ResNet [9] as our first approach because of the following reason:

- This model helps mitigate the vanishing gradient problem thanks to residual connections, so it can support training of very deep network, which captures more abstract features of the character in the image.
- The skip connections in ResNet allow for faster convergence during training, so it can facilitate with large-scale datasets.
- ResNet strongly supports transfer learning, so we can use pretrained models for the starting point of the training.

Based on the architecture introduced in the original paper, we decided to try 3 versions of ResNet: ResNet18, ResNet34, ResNet50.

Moreover, we adopted image augmentation techniques to improve the performance of the model. Specifically, we attempted to convert the original images, which had colored backgrounds, into a "black character, white background" format (BlackWhite version). This was achieved by using a fast non-local denoising method [1] in conjunction with adaptive thresholding. This approach allowed us to separate the characters from the background.

We experiment on two datasets: class-provided dataset and SuperNom.

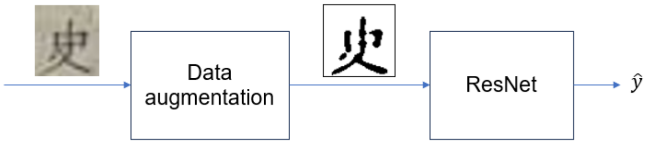


Figure 6: An illustration of BlackWhite pipeline

5.3 SimCLR

With the construction of above custom datasets, we have the idea of applying semi-supervised learning for better result. We adopt the SimCLR [3] as a pretrained model, then we finetune the received model to get a better result.

To be more specific, the training process is conducted as follows: For the pretrained SimCLR, we generate two augmented versions for each training image (both labeled and unlabeled): a random-rotation image and a color-jitter image. Due to resource limitations and the requirement for a large batch size to achieve better model improvement, we first convert all colored images to grayscale. The random-rotation image is created by rotating the image at a random angle, while the color-jitter image is produced by randomly adjusting the image's brightness and contrast. Following these augmentations, the two images are passed through a ResNet18 encoder to obtain their representations. These representations are then projected into a smaller latent space using a small neural network, such as an MLP. Finally, contrastive loss is applied to maximize the similarity between the two augmented images derived from the same original image, while minimizing the similarity between images from different originals:

$$\mathcal{L}_{\text{contrastive}} = -\log \frac{\exp(\cos(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp(\cos(z_i, z_k)/\tau)} \tag{1}$$

In the above loss function, z denotes the representation of an augmented image, z_i, z_j are two augmented images from the same image, $\cos(\cdot)$ represents the cosine similarity between two embeddings, τ is a hyper-parameter to control the strength of penalty on negative examples, N is the number of original images in a batch, $\mathbf{1}_{k \neq i}$ equals 1 when $k \neq i$, 0 when $k = i$.

In our finetuning model, we use labeled data, pass it through ResNet and neural network after pretraining, and add a dense layer so that the model can classify the data based on the given label.

The complete pipeline is illustrated in Figure 7.

We experiment on three datasets: class-provided dataset, SuperNom and SemiSuperNom.

6 RESULTS AND ANALYSIS

6.1 Results

6.1.1 *Extraction of Recognition Results.*

To extract recognition results from the model, the following steps were followed:

(1) **Preprocessing**: Each input image was preprocessed to match the dimensions and format expected by the model. This included resizing, normalization, and binarization.

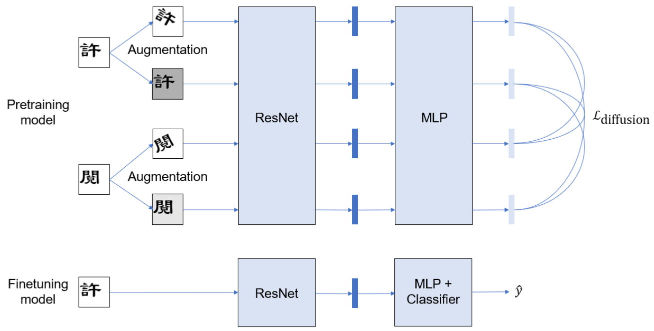


Figure 7: An illustration of SimCLR architecture

(2) **Model Inference**: The preprocessed image was fed into the model to obtain the predicted output. The model's architecture allowed for the extraction of the most probable class (character) for each image.

(3) **Result Storage**: The predicted results were stored in a structured format, including the original image, the predicted character, and the ground truth.

6.1.2 Experimental Settings. The model was implemented using the TensorFlow deep learning framework, which is written in Python. All experiments were conducted on the Kaggle platform, using two GPU T4 instances to ensure a fair comparison.

For the hyperparameters, the batch size was set to 1024. The temperature parameter τ for the contrastive loss function was explored within the range of $\{0.1, 0.2, 0.5, 1\}$. The maximum number of epochs was set to 200. An Inverse Time decay learning rate schedule was used, along with the Adam optimizer. Early stopping was employed, halting training if the validation accuracy did not improve for 15 consecutive epochs.

Model	Class-provided	SuperNom	SemiSuperNom
ResNet18	0.7042	0.6248	-
ResNet18 + BlackWhite	0.6335	-	-
ResNet34	0.6571	0.6439	-
ResNet34 + BlackWhite	0.6139	-	-
ResNet50	Overfitting		
SimCLR	0.8017	0.7636	0.7945

Table 2: Experimental results between our approaches with accuracy metric. Position with '-' sign means that we do not have results of the model on the dataset.

6.1.3 Overall Performance. The performance results for all the approaches are summarized in Table 2. We can easily observe that the best result is on SimCLR model with class-provided and SuperNom dataset. This result is aligned with the original paper, in which the incorporation of contrastive learning helps the model learn better representations.

BlackWhite variants exhibited lower accuracy compared to the original variants. This can be attributed to our observation that the image augmentation process, while successfully separating the characters from the background, also inadvertently removed some pixel information within the characters themselves. This loss of internal character detail made the images more challenging for the model to recognize accurately.

The class-provided dataset shows that the performance of the ResNet model with fewer layers is better than the performance of the ResNet model with more layers. However, a contradictory trend is observed with the SuperNom model. This discrepancy can be explained by the small size of the class-provided dataset. The smaller dataset size makes the smaller-layer ResNet model less prone to overfitting, resulting in better performance compared to the deeper ResNet model. In contrast, a larger dataset would allow the deeper ResNet model to learn more features of the characters, potentially leading to better performance.

The efficiency gained from having a larger dataset was not clearly evident when evaluating the models on the validation set using the SimCLR approach. Among all the datasets tested, SuperNom exhibited the worst performance. This can be attributed to the considerable number of noisy labeled images in this dataset, which made it challenging for the model to learn the true patterns of each character effectively. In contrast, the SemiSuperNom dataset, while not providing significant performance improvements, still yielded reasonably good results compared to the class-provided dataset. This can be explained by the fact that the noisy data was only used to differentiate the representations of different characters. As a result, after fine-tuning the model, the noise did not affect the performance as much as it did for the SuperNom dataset.

6.2 Analysis of Misrecognized Images

The misrecognized images were analyzed using the following criteria:

- **Character Similarity**: Comparison of the predicted character with the ground truth to identify visually similar characters.
- **Handwriting Style**: Examination of handwriting styles that the model struggled to recognize.
- **Image Quality**: Assessment of the quality of images, including factors like noise, resolution, and clarity.

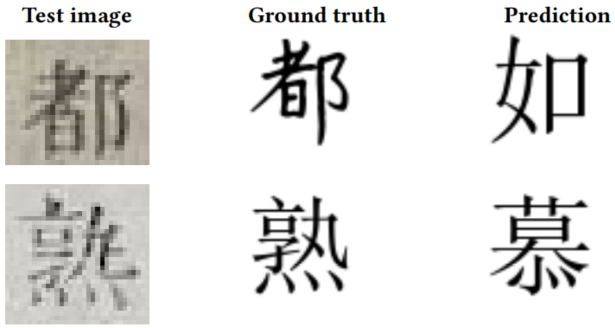


Figure 8: Comparison of test images, ground truth, and predictions that have visually similar patterns

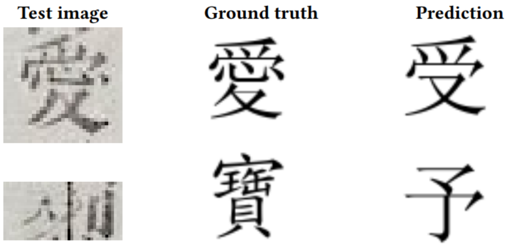


Figure 9: Comparison of test images, ground truth, and predictions that have bad quality

7 CONCLUSION

In conclusion, the recognition of Sino-Nom characters remains a critical task for the preservation and dissemination of cultural heritage. Our work has underscored the complexity of this challenge by analyzing two distinct datasets and experimenting with various methodologies. The results, with our best model achieving an accuracy of around 80%, represent a baseline in the automated recognition of these complex characters. However, there is still room for improvement. Future work could focus on refining the noise images when constructing datasets, applying better data augmentations, and integrating more sophisticated machine learning techniques, such as more advanced deep learning models and hybrid approaches, experimenting the effectiveness of our model when training for a longer time. Continued research in this area will not only contribute to the field of digital humanities but also support broader efforts in cultural preservation and historical scholarship in Vietnam.

REFERENCES

[1] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. 2011. Non-local means denoising. *Image Processing On Line*, 1, 208–212.

[2] Chun-Chieh Chang, Ashish Arora, Leibny Paola Garcia Perera, David Etter, Daniel Povey, and Sanjeev Khudanpur. 2019. Optical character recognition with chinese and korean character decomposition. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*. Vol. 5. IEEE, 134–139.

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

[4] Hoang-Quan Dang et al. 2022. Nomnaocr: the first dataset for optical character recognition on han-nom script. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 476–481.

[5] Trong Tuan Dao, Cong Thuong Le, Thi Duyen Ngo, and Thanh Ha Le. 2023. Detection and recognition of sino-nom characters on woodblock-printed images. In *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 1–6.

[6] Nadim Mahmud Dipu, Sifatul Alam Shohan, and KMA Salam. 2021. Bangla optical character recognition (ocr) using deep learning based image classification algorithms. In *2021 24th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 1–5.

[7] Manish Kumar Gupta, Surya Vikram, Siddharth Dhawan, and Ajai Kumar. 2023. Handwritten ocr for word in indic language using deep networks. In *2023 10th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 389–394.

[8] J Haritha, VT Balamurugan, KS Vairavel, N Ikram, M Janani, K Indrajith, et al. 2022. Cnn based character recognition and classification in tamil palm leaf manuscripts. In *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE, 1–6.

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

[10] Cristian Iorga and Victor-Emil Neagoe. 2019. A deep cnn approach with transfer learning for image recognition. In *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, 1–6.

[11] B Kalpana and M Hanmandlu. 2023. Offline handwritten basic telugu optical character recognition (ocr) using convolution neural networks (cnn). In *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 1195–1200.

[12] Abhishek Kashyap et al. 2022. Ocr of kannada characters using deep learning. In *2022 Trends in Electrical, Electronics, Computer Engineering Conference (TEECCON)*. IEEE, 35–38.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[14] Shilpa Kaur Manocha and Piyush Tewari. 2021. Comparative study of deep learning models for devanagari ocr. In *2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON)*. IEEE, 1–7.

[15] Cong Kha Nguyen, Cuong Tuan Nguyen, and Nakagawa Masaki. 2017. 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*, 37–41.

[16] Soham Patel, Dhyey Sanghavi, and Archana Nanade. 2023. Modernizing data processing: cnns and ocr for automated document classification and data extraction. In *2023 Global Conference on Information Technologies and Communications (GCITC)*. IEEE, 1–8.

[17] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

[19] Mingxing Tan and Quoc Le. 2019. Efficientnet: rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[20] Anh-Thu Tran, Ngoc-Yen Le, Dien Dinh, and Thai-Son Tran. 2021. Integrating nôm language model into nôm optical character recognition. In *International Conference on Artificial Intelligence and Big Data in Digital Era*. Springer, 47–60.

[21] Truyen Van Phan, Kha Cong Nguyen, and Masaki Nakagawa. 2015. A nom historical document recognition system for digital archiving. *International Journal on Document Analysis and Recognition (IJDAR)*, 19, 1, 49–64.