Application of AI in the Automatic Construction of a Sino-Nom – Vietnamese National Script Parallel Corpus

1st Phuc Bui Hong
University of Science, VNUHCM
Ho Chi Minh city, Vietnam
duyphuc2425@gmail.com

2nd Dinh Si Dien
CLC-Computational Linguistics Center, VNUHCM
Ho Chi Minh city, Vietnam
dinhsidien2008@gmail.com
*Corresponding author

3nd Nguyen Thuy Trang
University of Science, VNUHCM
Ho Chi Minh city, Vietnam
24C01023@student.hcmus.edu.vn

4rd Truong Thi Van Anh
University of Science, VNUHCM
Ho Chi Minh city, Vietnam
24C01002@student.hcmus.edu.vn

***Abstract*—Sino-Nom–Vietnamese parallel corpora are the primary training data for optical character recognition (OCR) and transliteration models on Sino-Nom texts. However, building such corpora remains difficult because OCR on Sino-Nom scripts is hindered by complex glyphs, degraded sources, and historical orthography. We present an AI-assisted method to automatically construct a Sino-Nom–Vietnamese parallel corpus by jointly exploiting semantic and visual similarity between Sino-Nom characters and their counterparts in Vietnamese (quốc ngữ). Concretely, we perform character-level alignment with Levenshtein distance, using the Vietnamese line as a semantic anchor to recover the most plausible Sino-Nom character that matches in both meaning and appearance. This procedure corrects noisy OCR outputs and elevates the quality of digitized classical texts. The resulting pipeline improves OCR post-processing and enables scalable corpus construction for recognition and transliteration of Sino-Nom texts.**

***Index Terms*—Artificial Intelligence (AI), Sino-Nom, Parallel Corpus, Levenshtein, Character Alignment, Natural Language Processing (NLP).**

I. Introduction

Sino-Nom is a logographic writing system historically used in Vietnam to record literary, administrative, and religious texts from the 11th to early 20th century. Despite its deep cultural value and linguistic richness, processing Sino-Nom materials in the digital era remains challenging. Most historical documents exist only in physical form or as degraded scans, while expertise in reading and interpreting the script is declining rapidly. At the same time, the Vietnamese national script, based on Latin characters with diacritics, has become the modern standard for written communication and education.

Constructing a high-quality parallel corpus between Sino-Nom and Vietnamese is essential for both cultural preservation and modern NLP, and—critically—constitutes the primary training data for OCR and transliteration on Sino-Nom materials. Such corpora support not only dictionary building and translation studies but also enable supervised learning for tasks like grapheme-to-phoneme modeling, transliteration, or historical document retrieval. However, the creation of this corpus is hindered by multiple challenges: (1) input images are often noisy, skewed, or low-resolution; (2) OCR for Sino-Nom remains error-prone due to visual similarity among thousands of characters; and (3) alignment between the Sino-Nom source and Vietnamese transliterations is typically not available and must be inferred.

A critical bottleneck is the acute shortage of curated training datasets for both OCR and Sino-Nom transliteration. Because most Vietnamese classical holdings survive only as images, any computational pipeline must first recognize characters; yet publicly available line- and character-aligned pairs are scarce, fragmented, inconsistently annotated, or restricted by institutional access. Building such resources manually demands expert paleographic knowledge, meticulous proofreading, and time-consuming alignment, while the number of qualified Hán Nôm readers is rapidly diminishing. At current rates, purely manual curation cannot scale—motivating an automated approach to bootstrap reliable supervision from noisy OCR outputs and Vietnamese transliterations.

To address these issues, we propose a novel and practical pipeline for automatically constructing a character-aligned Sino-Nom–Vietnamese parallel corpus. Our contributions include:

- A language-aware preprocessing stage that classifies image regions using OCR and applies YOLOv11-based line detection to crop meaningful text while removing decorative noise;
- A Sino-Nom correction algorithm combining visual similarity clusters and semantic alignment with Vietnamese transliterations using Levenshtein distance;
- A two-level alignment framework that operates at both line and character levels, accommodating OCR inconsistencies and misalignments using a dynamic matching window;
- A human-inspectable, color-coded evaluation protocol

to verify alignment quality, guiding error analysis and corpus refinement.

Beyond technical contributions, this work is driven by an urgent cultural need. A vast portion of Vietnamese history is documented solely in Sino-Nom, yet the number of experts capable of reading it is shrinking. Manual transliteration efforts, though accurate, are time-consuming and unable to keep pace with the rate of physical deterioration of ancient manuscripts. Without acceleration via automation, valuable cultural knowledge is at risk of being permanently lost.

Our pipeline provides a semi-automated solution that speeds up the digitization process while maintaining alignment accuracy. In the long term, such corpora can facilitate the development of AI systems—such as large language models (LLMs) and ChatGPT-style assistants—capable of understanding, querying, and translating Sino-Nom documents. By making these historical sources more accessible, we bridge the gap between traditional scholarship and modern computational tools, preserving cultural heritage for future generations.

II. Related Works

A. Levenshtein-based OCR Post-Correction for Historical Texts

Levenshtein distance remains one of the most reliable and interpretable measures for OCR post-correction. Srigiri and Saha [1] applied it in combination with word embeddings to correct OCR outputs for historical Hindi, reporting substantial improvements in character-level accuracy. Aditya Pal et al. [2] enhanced this approach by ranking candidates contextually using BERT embeddings and Levenshtein, achieving 81% accuracy on pre-modern texts. S Mihov et al. [3] introduced Levenshtein automata integrated with web-based dictionaries, optimizing both efficiency and precision. A recent survey by C Da et al. [4] highlights the adaptability and robustness of Levenshtein-based pipelines across multiple languages and historical scripts.

However, while these methods excel in Latin-based scripts, they struggle with ideographic languages with large character sets. Our work specifically addresses this by integrating visual similarity (glyph form clusters) and semantic anchoring from Vietnamese transliteration, tailored for the Sino-Nom script.

B. Google Vision OCR in Practical Applications

Google Vision OCR has emerged as a widely adopted solution for extracting textual data from scanned and degraded documents due to its support for multiple languages, ease of integration, and high accuracy in standard print environments. D Vaithiyanathan and M Muniraj [5] explored its use in assistive reading applications for visually impaired users, highlighting its ability to handle low-quality text in real-time scenarios. In a large-scale industrial setting, R Arief et al. [6] embedded Google Vision OCR into a Hadoop-based document processing pipeline, demonstrating its scalability and reliability for high-throughput workloads.

Moreover, NPT Prakisya et al. [7] conducted a comparative evaluation between Google Vision and open-source OCR engines such as Tesseract, showing that Google Vision consistently outperformed alternatives in both character-level and word-level accuracy—especially on documents with faded ink, skewed layouts, and historical typography. These findings confirm the robustness of Google Vision OCR in multilingual and real-world use cases, justifying its role in our pipeline for processing Vietnamese national script.

The original documents often contain a mixture of Sino-Nom and Vietnamese script, which presents challenges for automated processing. We trained a YOLOv11 [8] object detection model to automatically crop regions of interest, facilitating effective segmentation and noise reduction.

C. Sino-Nom OCR Using Kim Hán Nôm

Given the outstanding advantages of Google Vision, we continue to use this tool for Vietnamese OCR; however, general OCR engines like Tesseract and Google Vision perform poorly on Sino-Nom due to limited training on ideographic glyphs. Sino-Nom scripts pose unique challenges for OCR due to their ideographic structure and historical variability. Standard OCR systems like Google Vision and Tesseract are generally not trained on these character sets. To address this, we integrate Kim Hán Nôm, an OCR system developed by the Computational Linguistics Center (CLC Lab) [9], specifically designed for recognizing classical Vietnamese characters. Trained on annotated corpora of stelae and manuscripts, Kim Hán Nôm integrates deep learning with stroke normalization and layout correction, achieving high accuracy in both printed and handwritten forms. Its utility extends to digital humanities, historical linguistics, and cultural preservation.

These studies collectively illustrate the progress in OCR extraction and post-correction for multilingual and historical documents. Our approach integrates general-purpose tools like Google Vision and YOLOv11 with specialized OCR systems such as Kim Hán Nôm, tailored for Sino-Nom scripts. Importantly, Levenshtein distance emerges as a core algorithm that remains highly relevant and effective in modern post-correction pipelines, particularly when combined with embedding-based or contextual models for ranking and filtering.

III. Methodology

The primary goal of this chapter is to present a structured approach for automatically constructing a high-quality parallel corpus between Sino-Nom texts and their Vietnamese national script translations. The methodology integrates OCR with noise reduction, script-aware segmentation, character-level correction based on Levenshtein distance, and automatic alignment techniques. It is designed to handle the unique challenges of historical documents, including visual degradation, multilingual script mixing, and inconsistent formatting. The full pipeline encompasses scan preprocessing, OCR processing, language-based filtering, semantic-aware correction, alignment, and corpus validation.

A. Preprocessing for Scan Photo

The first stage in constructing the Sino-Nom – Vietnamese parallel corpus involves preprocessing scanned historical documents to optimize them for OCR and subsequent text extraction. As shown in Fig. 1, this step includes language-aware processing and noise reduction through image cropping.

The first stage in constructing the Sino-Nom – Vietnamese parallel corpus involves preprocessing scanned historical documents to optimize them for OCR and subsequent text extraction. As shown in Fig. 1, this step includes both language-aware filtering and image denoising through region-based cropping.

We begin by applying the OCR capabilities of Google Vision to extract raw textual content from scanned PDF images. This tool supports multilingual scripts and provides bounding box information that can assist in later spatial structuring. However, due to the historical nature of the source documents, each scanned page often contains a mixture of Sino-Nom characters and Vietnamese national script, along with decorative frames, calligraphic headers, and marginal annotations that introduce significant noise (figure 2). In many cases, OCR models misinterpret these visual elements—especially borders and table structures—as actual characters, distorting the output and complicating subsequent processing steps.

To address this challenge, we employed a two-stage filtering mechanism. First, we used the `langdetect` library to perform language classification on each OCR output segment. This step enables us to separate Sino-Nom and Vietnamese script blocks, allowing for tailored handling of each stream. Such separation is critical given the distinct tokenization rules, semantic structures, and error characteristics between the two scripts.

Second, we trained a YOLOv11 object detection model to automatically crop regions of interest from each scanned page. The model was trained on 600 manually annotated samples, each labeled to detect only the textual regions while ignoring non-informative elements such as ornate frames, floral decorations, or blank margins. This cropping process plays a pivotal role in reducing OCR noise. By excluding visual clutter, we prevent OCR engines from erroneously recognizing these artifacts as valid characters — a particularly common issue in historical Sino-Nom manuscripts, where framing elements can resemble stroke-dense ideographs.

In effect, YOLOv11 serves as a denoising gate that enhances OCR reliability for both script streams. Unlike traditional image pre-processing (e.g., thresholding or blurring), this object-detection approach enables spatially precise, content-aware filtering tailored to complex layouts. As a result, the output of this stage is a cleaner set of text segments—separated by script and stripped of visual noise—ready for OCR and semantic correction in the next phase.

B. OCR for Raw Text

Following the cropping and noise removal stage, we apply Optical Character Recognition (OCR) to extract raw text from both Sino-Nom and Vietnamese regions. Given the distinct script and linguistic properties of each, we apply different OCR tools tailored to their respective characteristics.

For Sino-Nom text regions, we employ KimHanNom, a specialized OCR engine developed by the Computational Linguistics Center (CLC Lab). Unlike generic OCR systems, KimHanNom is explicitly designed to handle the complex visual structure of classical Sino-Nom scripts. Its character set includes thousands of traditional and variant glyphs, enabling more accurate recognition of historical documents written in Sino-Nom. For Vietnamese regions, we continue to use Google Vision due to its superior performance on Latin-based scripts, especially in modern and well-printed formats. No post-processing was applied to the Vietnamese OCR stream, as recognition accuracy in this domain was sufficiently high.

However, even with a domain-specific engine like KimHanNom, the recognition of Sino-Nom characters remains susceptible to errors—particularly when source documents contain degraded ink, uncommon glyph variants, or closely overlapping strokes. To mitigate these issues, we developed a character correction framework based on Levenshtein distance, leveraging curated visual and semantic associations across characters.

Correction Algorithm.

Given an OCR-recognized character c' from KimHanNom, we execute the following correction pipeline:

1) Visual Similarity Filtering: Retrieve a list of visually similar Sino-Nom characters to c', using a handcrafted dataset in which each character is linked to up to 20 lookalike forms (based on radical overlap or structural proximity).
2) Use the aligned Vietnamese transliteration to filter the candidate set. Specifically, we map Vietnamese syllables (e.g., “thọ”) to their known Sino-Nom equivalents (e.g.,壽) using a bilingual dictionary built from historical lexicons.
3) Set Intersection:
   - If the intersection between the visually similar set and the semantically plausible set is empty, apply Levenshtein distance between c' and each visually similar candidate to select the nearest character.
   - If exactly one character is found in the intersection, it is used as the correction.
   - If multiple characters remain, we again apply Levenshtein distance within the intersection to choose the best match.

This approach allows us to recover from recognition errors by jointly leveraging visual structure similarity (form-based filtering) and semantic alignment (meaning-based disambiguation). By using Vietnamese national script as a semantic anchor, we are able to eliminate implausible candidates and enhance correction precision.

This correction mechanism forms a central component of our pipeline, substantially improving the Sino-Nom OCR stream’s fidelity and enabling reliable downstream alignment.

Correction Dataset. This algorithm is powered by two structured resources:

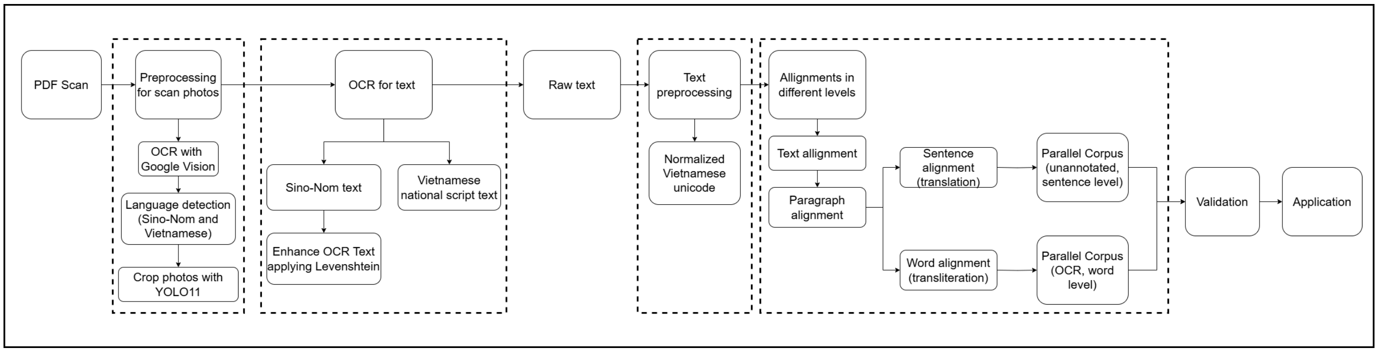


Figure 1. Overall pipeline for constructing a Sino-Nom – Vietnamese parallel corpus.

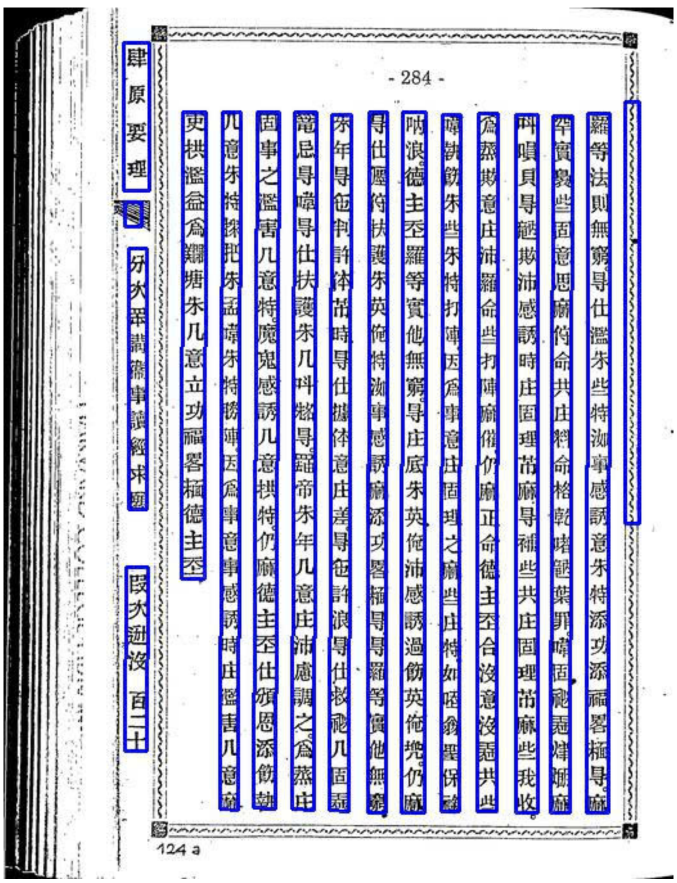
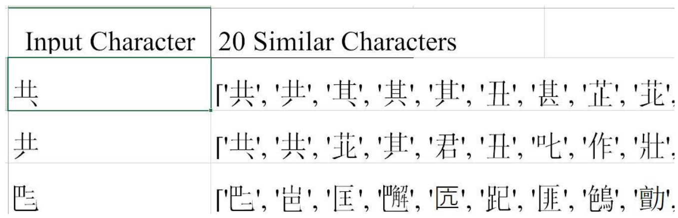


Figure 2. Example for OCR with full page

- A curated list of 26,044 Sino-Nom characters, each annotated with up to 20 visually similar glyphs. These were derived from traditional dictionaries and expert annotation. Figure 3 shows a sample of the dataset.
- A bilingual Vietnamese–Sino-Nom mapping dataset, compiled from multiple annotated corpora and historical references.

C. Text Preprocessing

After the OCR and character correction stages, the extracted textual data is organized into three distinct components:



Input Character	20 Similar Characters
共	['共', '井', '其', '其', '其', '丑', '甚', '芷', '茈',
井	['共', '共', '茈', '其', '君', '丑', '叱', '作', '壯',
巴	['巴', '岜', '匡', '𨇾', '匹', '跁', '匪', '鴝', '勯',

Figure 3. Sino-Nom similar visual dataset example

(1) the phonetic transliteration in Vietnamese script, (2) the original Sino-Nom text, and (3) the bounding box metadata corresponding to the Sino-Nom segments. Each component undergoes a dedicated preprocessing routine to prepare the data for alignment and corpus construction. While the system provides a generalizable and reproducible pipeline, certain steps remain adaptable to the structural variations commonly found in historical documents.

Vietnamese Transliteration Preprocessing. The transliterated Vietnamese layer requires careful normalization due to the complexity of the script's diacritic system and Unicode variability. Characters that appear visually identical may differ in their underlying Unicode representations, potentially causing misalignment. Accordingly, we normalize all Vietnamese text to Unicode NFC to ensure corpus-wide consistency.

Additional preprocessing steps include:

- **Removal of noisy characters:** Extra whitespace, malformed OCR tokens, and non-standard symbols are filtered out.
- **Masking of presentation elements:** Structural components such as headers, speaker annotations, or document-specific markers are temporarily masked to prevent interference with downstream parsing.
- **Punctuation removal:** All punctuation marks are stripped to yield a clean, phoneme-like sequence suitable for alignment.
- **Number normalization:** Numerical expressions (e.g., "123") are converted into their full Vietnamese verbal equivalents (e.g., "một trăm hai mươi ba") using a deterministic mapping module.
- **Preservation of phonetic content:** Only semantically

relevant content is retained, enabling precise alignment with Sino-Nom characters.

The final transliteration stream is preserved as a continuous character sequence without sentence segmentation, as its primary role is to facilitate grapheme-to-phoneme modeling and character-level alignment.

Sino-Nom Text Handling. No additional preprocessing is performed on the Sino-Nom text after the character correction phase. The output from the KimHanNom OCR engine—refined using the Levenshtein-based correction algorithm—is considered final and directly usable for alignment. The text and its associated bounding boxes are preserved without modification to retain spatial fidelity and document structure.

Adaptability to Document Variants. While the preprocessing procedures above define a consistent and modular framework, the layout and formatting of historical texts are often highly variable. Depending on the document genre—imperial edicts, annotated records, petitions, or official decrees—further customization may be required. The pipeline is therefore designed to serve as a supportive baseline, allowing researchers to integrate domain-specific rules and manual refinements as needed. This balance of automation and flexibility ensures the pipeline can scale across a diverse range of classical sources while maintaining high alignment accuracy.

D. Alignment at Different Levels

The final stage of the pipeline constructs a transliteration-based parallel corpus by aligning Sino-Nom text with its corresponding phonetic rendering in Vietnamese. The alignment process operates at two levels of granularity: bounding box-level alignment and character-level alignment within each box. Unlike typical sentence- or paragraph-level alignment in bilingual corpora, this setting is character-centric and tightly constrained by the structure of the original scan layout.

Bounding Box Indexing. Prior to alignment, each OCR segment (bounding box) is assigned a unique index to maintain its spatial order. This index is computed based on the book title, page number, and the reading order of bounding boxes after spatial sorting (typically top-to-bottom, left-to-right). This indexing scheme is implemented during the OCR or preprocessing stage and ensures that segments from the Sino-Nom OCR output and the phonetic transliteration stream can be matched deterministically.

Box-Level Alignment. For each indexed bounding box, the Sino-Nom text extracted from OCR is paired with its corresponding transliteration segment. The assumption is that each box contains a linear sequence of Sino-Nom characters, and the transliteration provides a phonetic rendering in 1-to-1 character order. Therefore, we begin by aligning one Sino-Nom character to one Vietnamese syllable within the same bounding box.

Character-Level Alignment with Levenshtein Adjustment. In practice, the number of Sino-Nom characters in a box may not perfectly match the number of syllables in the transliteration due to OCR noise or slight formatting shifts. To handle this, we apply a flexible alignment strategy:

- Let n be the number of Sino-Nom characters in the bounding box.
- We allow a window of $n \pm k$ Vietnamese syllables (typically $k = 1, .., 5$) to be considered for alignment.
- We use Levenshtein algorithm at the character level to align Sino-Nom glyphs with the most plausible phonetic units in the Vietnamese stream.

If a character from the Sino-Nom OCR output has no suitable match in the transliteration window, an underscore "_" is inserted as a placeholder to preserve alignment structure. This ensures that each line of the corpus maintains 1-to-1 alignment, even in the presence of missing or noisy data.

Algorithm 1 LEVENSHTEINALIGNBOXES

Input: $\mathcal{N} = \langle n_1, \ldots, n_m \rangle$ (Sino–Nom), $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ (Vietnamese), `similar_df`, `trans_df`
Output: Aligned pair $(\texttt{AlignedN}, \texttt{AlignedQ})$

$D_{\text{trans}}, D_{\text{sim}} \leftarrow \texttt{BuildDicts}(\texttt{trans\_df}, \texttt{similar\_df})$
$m \leftarrow |\mathcal{N}|$, $n \leftarrow |\mathcal{Q}|$ $dp \in \mathbb{N}_0^{(m+1)\times(n+1)}$; $bt \in \{\uparrow, \leftarrow, \nwarrow, \varnothing\}^{(m+1)\times(n+1)}$
Initialize boundaries for $i \leftarrow 0$ **to** m **do**
  $dp[i, 0] \leftarrow i$; $bt[i, 0] \leftarrow \uparrow$
for $j \leftarrow 0$ **to** n **do**
  $dp[0, j] \leftarrow j$; $bt[0, j] \leftarrow \leftarrow$
$bt[0, 0] \leftarrow \varnothing$
Dynamic programming for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** n **do**
    $cost \leftarrow \big(0$ if $\texttt{Compatible}(n_i, q_j, D_{\text{trans}}, D_{\text{sim}})$ else $1\big)$
    $u \leftarrow dp[i-1, j] + 1$; $l \leftarrow dp[i, j-1] + 1$;
    $d \leftarrow dp[i-1, j-1] + cost$ $(dp[i, j], bt[i, j]) \leftarrow$
    $\arg\min\{(u, \uparrow), (l, \leftarrow), (d, \nwarrow)\}$
Backtrace $\texttt{AlignedN} \leftarrow [\,]$, $\texttt{AlignedQ} \leftarrow [\,]$, $i \leftarrow m$, $j \leftarrow n$ **while** $i > 0$ **or** $j > 0$ **do**
  if $i > 0$ **and** $j > 0$ **and** $bt[i, j] = \nwarrow$ **then**
    $\texttt{push\_front}(\texttt{AlignedN}, n_i)$;
    $\texttt{push\_front}(\texttt{AlignedQ}, q_j)$; $i \leftarrow i - 1$;
    $j \leftarrow j - 1$
  else if $i > 0$ **and** $bt[i, j] = \uparrow$ **then**
    $\texttt{push\_front}(\texttt{AlignedN}, n_i)$;
    $\texttt{push\_front}(\texttt{AlignedQ}, *)$; $i \leftarrow i - 1$
  else
    $\texttt{push\_front}(\texttt{AlignedN}, *)$;
    $\texttt{push\_front}(\texttt{AlignedQ}, q_j)$; $j \leftarrow j - 1$
return $(\texttt{AlignedN}, \texttt{AlignedQ})$
Function $\texttt{Compatible}(n, q, D_{trans}, D_{sim})$**:**
  return $(q \in D_{\text{trans}}[n])$ **or** $(\exists c \in D_{\text{sim}}[n] : q \in D_{\text{trans}}[c])$

Final Output Format.

The result is a character-aligned transliteration corpus as showed in figure 4, where each line corresponds to a single bounding box. Each line contains a pair of aligned sequences:

Img_Box_ID	Img_Box_Coordinate	SinoNom_OCR	SinoNom_Char	ChuQN_txt
HVK_001.001.067.002.jpg	[[128, 77], [317, 71], [318, 99], [129, 105]]	焌唏祐焆動悉霝霔	焌唏祐焆動悉霝霔	Bén hơi rơm lửa động phòng mưa mây
HVK_001.001.067.003.jpg	[[150, 121], [301, 118], [302, 145], [151, 148]]	捤渃招物巾秭	捤渃招物巾秭	Vẩy nước chậu vắt khăn tay
HVK_001.001.067.004.jpg	[[129, 150], [320, 148], [320, 176], [129, 178]]	欺䀡踏蓓欺旴𡎢鍾	欺䀡踏蓓欺旴𡎢鍾	Khi đêm đạp bóng khi ngày ngồi chung
HVK_001.001.067.005.jpg	[[144, 179], [306, 177], [307, 205], [144, 207]]	花桃㐌憚𩙍東	花桃㐌憚𩙍東	Hoa đào đã dạn gió đông

Figure 4. Overall pipeline for constructing a Sino-Nom – Vietnamese parallel corpus.

the corrected Sino-Nom characters and their aligned Vietnamese phonetic forms. This format enables both linguistic analysis (e.g., grapheme-to-phoneme modeling) and potential supervised training for future Sino-Nom OCR or transliteration system.

IV. Experiments

A. Experimental Setup

To evaluate the effectiveness of the proposed pipeline, we conducted experiments on three classical Vietnamese texts, each featuring Sino-Nom characters and corresponding transliterations in the Vietnamese national script:

- **Lives of the Saints (Sách Truyện Các Thánh)**
- **Essentials of Christian Doctrine (Tư Nguyên Yếu Lý)**
- **The History of Dai Nam in Verse (Đại Nam Quốc Sử Diễn Ca)**

All materials were sourced from publicly available online repositories. As such, the raw data consisted of low-resolution scanned images exhibiting various forms of degradation, including compression artifacts, uneven lighting, misaligned pages, and ink fading. This challenging input condition highlights the practical value and robustness of our preprocessing and correction pipeline, particularly in scenarios where high-quality digitization is not available.

The images were first classified into two categories—Sino-Nom and Vietnamese—using Google Vision's built-in language detection. Each group was then processed by a script-specific YOLOv11 model previously fine-tuned to detect line-level text regions in the respective writing system. The resulting bounding boxes (figure 5 were indexed by document metadata (book title, page number, and bounding box order) to preserve reading order and ensure deterministic alignment.

Text recognition was performed using two complementary OCR tools:

- **KimHanNom** for Sino-Nom characters, leveraging its specialized support for traditional glyph variants and historical typography.
- **Google Vision** for Vietnamese text, optimized for modern Latin-based scripts.

After OCR, character-level alignment was applied within each bounding box using the Levenshtein distance algorithm. A flexible alignment window of $n \pm k$ character window (typically $k = 5$) was used to handle minor inconsistencies in character count between the Sino-Nom and transliteration streams. Missing or unaligned characters were explicitly marked using an underscore ("_"), preserving alignment structure for further analysis. To assist human evaluation, we applied a color-coded visualization scheme (black, blue, red) to represent alignment confidence levels.

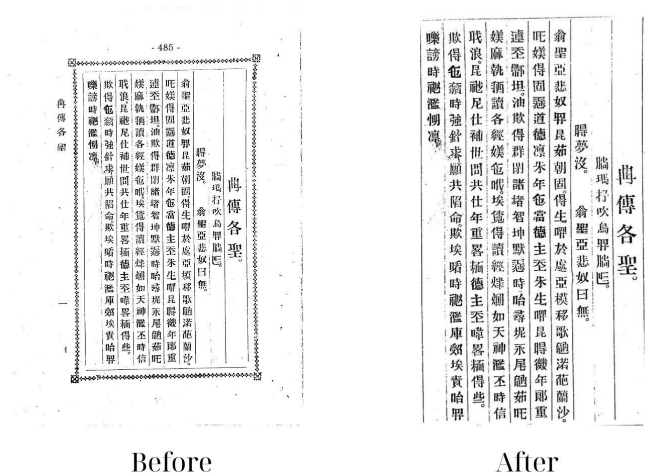


Figure 5. Result of crop image with YOLOv11

B. Evaluation Metrics

Color assignment rules:

- **Black**: OCR character exactly matches the dictionary `QuocNgu_SinoNom_Dic`.
- **Blue**: OCR character is approximately correct, found in `SinoNom_Similar_Dic_v2` with high Unicode similarity.
- **Red**: Incorrect character, not found in either dictionary.

Metric calculation:

Let:

- B: number of black (correct) characters
- G: number of blue (approximate match) characters
- R: number of red (incorrect) characters
- $T = B + G + R$: total number of evaluated characters

$$\text{Accuracy} = \frac{B+G}{T}, \quad \text{Red Ratio} = \frac{R}{T} \quad \text{Green Ratio} = \frac{G}{T}$$

$$\text{Green Ratio} = \frac{B}{T}$$

C. Results

Table I
CHARACTER-LEVEL ALIGNMENT EVALUATION

Book Title	Red (%)	Blue (%)	Black (%)	Accuracy (%)
Sách truyện các thánh	15%	2%	83%	85%
Tư Nguyên Yếu Lý	20%	1%	79%	80%
Đại Nam Quốc Sử Diễn Ca	10%	1%	89%	90%

D. Analysis

The results demonstrate the robustness of the proposed pipeline:

- **High Black ratio (79–89%):** A majority of characters were correctly aligned one-to-one, indicating that the bounding box indexing and character-level alignment processes are both reliable.
- **Strong overall accuracy (80–90%):** Verified correctness of aligned character pairs shows that the combined OCR and alignment system is practical for real-world digitization tasks.
- **Low Blue ratio (<2%):** In cases where multiple candidate characters were found, the use of Levenshtein distance successfully selected the most plausible match.
- **Decreasing Red ratio:** The rate of unmatched characters (Red) decreased as the character reference library and alignment logic improved, indicating scalability of the matching system.

The coloring strategy not only enhanced transparency during post-alignment review but also provided a valuable visual aid for error analysis and future corpus refinement.

The results demonstrate the robustness of the proposed pipeline:

The coloring strategy not only enhanced transparency during post-alignment review but also provided a valuable visual aid for error analysis and future corpus refinement.

V. LIMITATION

Each processing step in our pipeline requires manual verification to ensure optimal results. Although automated cropping using the detection model achieves high accuracy in most cases, some bounding boxes are still incorrectly predicted, leading to incomplete or misaligned cropped images. These errors, albeit limited in number, must be manually corrected to preserve the integrity of subsequent steps.

Additionally, the alignment stage is particularly sensitive to missing or shifted bounding boxes. Even a single undetected region can propagate alignment errors, affecting the layout of an entire page. Therefore, it is essential to manually review and adjust results after each major step—especially cropping and alignment—to maintain high overall quality in the final OCR output.

VI. CONCLUSION

This paper presents a comprehensive pipeline for constructing a character-aligned parallel corpus between Sino-Nom texts and their Vietnamese transliterations. Our method addresses key challenges in historical document digitization—including image noise, OCR inaccuracies, and alignment complexity—through a combination of language-aware preprocessing, visual-semantic character correction, and Levenshtein-based alignment. The pipeline has been evaluated on real-world low-quality scanned texts, achieving over 80% character-level alignment accuracy across multiple books.

Beyond the technical results, this work contributes to the broader mission of preserving Vietnamese cultural heritage. As expertise in reading Sino-Nom declines and physical documents continue to deteriorate, automated tools become increasingly necessary to accelerate digitization and transliteration. Our approach not only supports current linguistic and historical research but also lays the groundwork for future integration with large language models (LLMs) and other AI applications in digital humanities.

In future work, we aim to expand the corpus coverage, incorporate more sophisticated neural-based correction models, and explore sentence- or paragraph-level alignment for full translation tasks. We also plan to release annotated corpora and tools to support further research in Sino-Nom processing and historical language understanding.

VII. ACKNOWLEDGEMENTS

The research is financially supported by Ho Chi Minh City Department of Science and Technology.

REFERENCES

[1] S. Srigiri and S. K. Saha, "Spelling correction of ocr-generated hindi text using word embedding and levenshtein distance," in *Nanoelectronics, Circuits and Communication Systems*, V. Nath and J. K. Mandal, Eds. Singapore: Springer Singapore, 2020, pp. 415–424.

[2] A. Pal and A. Mustafi, "Vartani spellcheck - automatic context-sensitive spelling correction of ocr-generated hindi text using BERT and levenshtein distance," *CoRR*, vol. abs/2012.07652, 2020. [Online]. Available: https://arxiv.org/abs/2012.07652

[3] S. Mihov, S. Koeva, C. Ringlstetter, K. U. Schulz, and C. Strohmaier, "Precise and efficient text correction using levenshtein automata, dynamic web dictionaries and optimized correction models," in *Proceedings of Workshop on International Proofing Tools and Language Technologies*, 2004.

[4] C. Da, P. Wang, and C. Yao, "Levenshtein ocr," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022, pp. 322–338.

[5] D. Vaithiyanathan and M. Muniraj, "Cloud based text extraction using google cloud vison for visually impaired applications," in *2019 11th international conference on advanced computing (ICoAC)*. IEEE, 2019, pp. 90–96.

[6] R. Arief, A. B. Mutiara, T. M. Kusuma *et al.*, "Automated extraction of large scale scanned document images using google vision ocr in apache hadoop environment," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, 2018.

[7] N. P. T. Prakisya, B. T. Kusmanto, and P. Hatta, "Comparative analysis of google vision ocr with tesseract on newspaper text recognition," *Media of Computer Science*, vol. 1, no. 1, pp. 31–46, 2024.

[8] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics yolo," 2023, if you use this software, please cite it using the metadata from this file. Available at: https://github.com/ultralytics/ultralytics. [Online]. Available: https://ultralytics.com

[9] CLC Lab, "Kim hán nôm – automatic transliteration tool from sino-nôm to vietnamese script," 2022, online. [Online]. Available: https://tools.clc.hcmus.edu.vn