A Hybrid Deep Learning Approach to Keyword Spotting in Vietnamese Stele Images

Anna Scius-Bertrand[1], Marc Bui[2] and Andreas Fischer[1]
[1]University of Fribourg and HES-SO, Fribourg, Switzerland
[2]Ecole Pratique des Hautes Etudes, Paris, France
E-mail: anna.scius-bertrand@unifr.ch, marc.bui@ephe.psl.eu, andreas.fischer@unifr.ch

Keywords: hybrid deep learning, document image analysis, keyword spotting, annotation-free, Hausdorff edit distance, Vietnamese steles, Chu Nom

Received: April 5, 2023

In order to access the rich cultural heritage conveyed in Vietnamese steles, automatic reading of stone engravings would be a great support for historians, who are analyzing tens of thousands of stele images. Approaching the challenging problem with deep learning alone is difficult because the data-driven models require large representative datasets with expert human annotations, which are not available for the steles and costly to obtain. In this article, we present a hybrid approach to spot keywords in stele images that combines data-driven deep learning with knowledge-based structural modeling and matching of Chu Nom characters. The main advantage of the proposed method is that it is annotation-free, i.e. no human data annotation is required. In an experimental evaluation, we demonstrate that keywords can be successfully spotted with a mean average precision of more than 70% *when a single engraving style is considered.*

Povzetek: Predstavljen je hibridni pristop za iskanje ključnih besed v slikah z nagrobnikov, ki združuje globoko učenje in strukturno modeliranje Chu Nom znakov. Ključne besede so uspešno prepoznane s povprečno natančnostjo več kot 70%.

1 Introduction

Vietnamese steles are of great value for historians, as the stone engravings are a unique source of information to understand the social, economic, and belief structures in the villages. The Vietnamica[1] project, in particular, aims to investigate pious donations from ordinary people offered to local shrines. For this purpose, about 40,000 digital stele images are studied, which may contain hundreds of Chu Nom characters on a single image. To cope with this vast amount of characters, automatic image analysis methods are needed that are able to transcribe the contents of the steles into machine-readable form for searching and browsing. Some examples of stele images are shown in Figure 1, highlighting significant differences in column layout and image quality across different steles.

Although the state of the art for handwriting recognition for historical documents has made great progress in the past decades, it remains a difficult problem and an active field of research [1]. Keyword spotting [2] has been proposed early on as an alternative to automatic transcription for difficult cases, where a full transcription is not feasible with high accuracy. The goal is to identify specific search terms of interest, either by providing a template image of the keyword (*query-by-example*) or by providing a textual representation of the search term (*query-by-string*).

Similar to developments in other related fields, such as computer vision and natural language processing, the different methods for keyword spotting can be divided into three groups, namely *heuristic* methods, *machine learning based* methods, and *deep learning based* methods. They are roughly ordered by time, heuristic methods being the oldest, but they still coexist and new approaches are being developed for all three groups.

Heuristic methods incorporate domain knowledge about the handwriting and are able to match images directly, i.e. keyword images and images from the manuscript, in order to retrieve similar instances in a query-by-example scenario. Early examples include dynamic time warping methods based on contour features [3] and gradient features [4], as well as segmentation-free methods based on scale-invariant feature transform (SIFT) [5]. More recently, a graph-based approach has been proposed in [6], which relies on a structural representation of the handwriting and uses an approximate graph edit distance to compare handwriting graphs.

Machine learning methods pursue the paradigm of learning by example. They train keyword models with the help of learning samples, i.e. manually annotated handwriting images. In a first step, characteristic features are manually defined based on domain knowledge and in a second step, different machine learning methods are used to learn keyword models based on the features. Examples of this group of keyword spotting methods include methods based on hidden Markov models (HMM) with geometric

[1] https://vietnamica.hypotheses.org

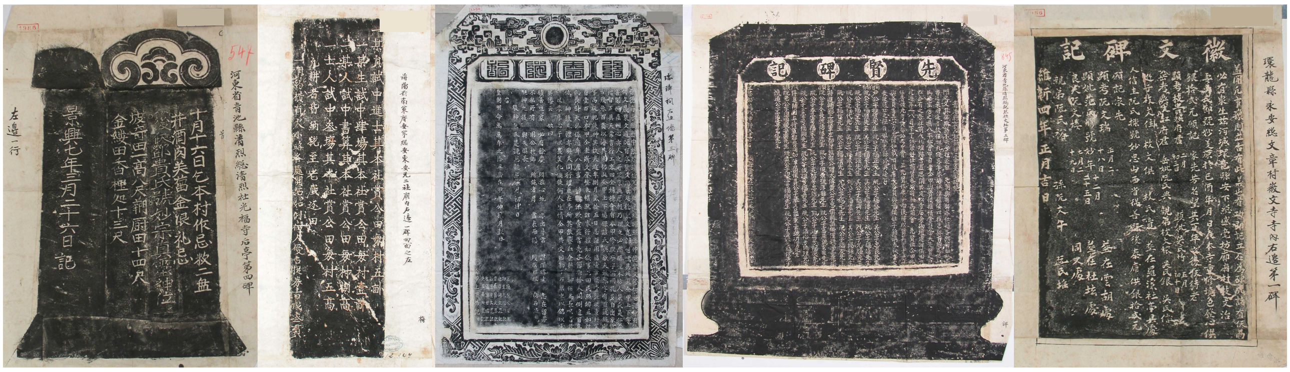


Figure 1: Example stele images.

features [7] and bag of local features [8], as well as bidirectional long short-term memory networks (BLSTM) with geometric features [9]. After training keyword models, the user can perform a query-by-string search, without the need of providing example images of the keyword.

Finally, deep learning methods are also based on the learning-by-example paradigm but they do not require manually defined features. Instead, they aim at learning characteristic representations, so-called embedding spaces, automatically from the data. Images as well as textual representations can be embedded in the same space, such that both query-by-example and query-by-string can be realized. A prominent example is the PHOCNet [10], which learns an embedding space based on pyramidal histogram of characters (PHOC) representations.

Today, the best keyword spotting performance is achieved by means of deep learning methods. However, they require a considerable amount of manually annotated training samples. In the case of historical Vietnamese steles, such learning samples can only be provided by experts, who are able to read the ancient Chu Nom script. It is thus time-consuming and costly to build a comprehensive training dataset, which is representative for the heterogeneous collection of stele images (see Figure 1). At the time of the writing, such a training set is not available for the 40,000 stele images.

In this article, we present a hybrid deep learning method for keyword spotting in historical Vietnamese stele images. It aims to combine deep learning with heuristic methods, such that the domain knowledge of the heuristic methods can compensate the lack of annotated learning samples. Indeed, it is an *annotation-free* method that does not require any human annotations at all.

The proposed method can be applied directly to the original stele images and consists of two processing steps. First, characters are detected using deep neural networks that are trained on synthetic images with printed Chu Nom characters and then auto-calibrated to real stele images. Secondly, the structure of the Chu Nom characters is modeled with a graph-based representation and matched with search terms using an approximate graph edit distance, in order to efficiently perform query-by-example keyword spotting. A comprehensive experimental evaluation is performed to measure the spotting performance.

The remainder of this article is structured as follows. Section 2 discusses related work on stele images, Section 3 provides more details about the content of the steles and the image acquisition, Section 4 presents the proposed keyword spotting method, and Section 5 details the experiments. Finally, conclusions are drawn in Section 6.

2 Related work

Initial work on the stele images has focused on the task of layout analysis with the aim to segment stele images into columns and characters. Such an initial segmentation is an important preprocessing step for character recognition. Notable work in this domain includes [11], where a heuristic method based on Voronoi diagrams is proposed to segment characters, and [12], where a deep learning approach based on semantic segmentation is pursued to detect columns with only a small number of human annotations. Recently, in [13] a deep learning method based on object detection networks has been introduced for character segmentation, which does not require human annotations and generalizes well to different layouts and engraving styles. In [14], a generative deep learning model has been suggested to create synthetic Chu Nom characters in different engraving styles. Recent work on Chu Nom also includes the U-Net based approach reported in [15], which was studied in the context of manuscripts.

Our method builds upon the character segmentation method of [13] and goes a step further to perform keyword spotting. For graph-based character modeling we rely on keypoint graphs, which have been studied for Latin scripts [6] and Chu Nom characters in manuscripts [16] before, but not for stone engravings. Important adaptations to the logographic writing system include the use of super-resolution to better model small strokes that distinguish similar Chu Nom characters. The graphs are efficiently matched using the Hausdorff edit distance [17], an approximation of graph edit distance that can be calculated in quadratic time with respect to the number of graph nodes.

Efficient graph matching is especially important in the context of super-resolution when Chu Nom graphs may contain over 100 nodes.

Preliminary results have already been published in a conference paper [18]. In this article, we provide a more detailed description of the hybrid deep learning method and significantly extend the experimental evaluation. Instead of considering only 8 steles, we conduct more comprehensive experiments on 20 stele images with manual ground truth. Furthermore, we study the important case of spotting keywords within the same style of engravings and compare it to a scenario with mixed styles. This study serves the purpose to better understand the possibilities and limitations of the proposed method.

3 Dataset

The 40,000 stele images represent about 25,000 steles, i.e. man-sized stones with engravings, which were erected in Vietnamese villages between the 16th and the 20th century. The majority of the steles represent donations made by villagers to the local shrines and are engraved in the ancient Vietnamese Chu Nom writing system [19]. However, they can also contain information about finances, constructions, and demarcations, thus informing about the social, economic, and religious life of the villages. The steles were erected for all to see and were able to withstand adverse weather conditions and armed conflicts. Nevertheless, they may contain degradations, fissures, and impacts, which may render parts of the steles illegible (see Figure 1).

The images of the steles were obtained by the French School of the Far East (EFEO) and the Institute of Han Nom Studies by means of stampings [19]. A sheet of paper is pressed on the stone and fixed with a binder, e.g. banana juice. Then, ink is applied with a roller on the paper over the entire surface of the stele, such that engravings appear in white and the stone background, as well as characters written in relief, appear dark in the color of the ink. Finally, the paper is photographed to obtain digital stele images. Such pictures of the stampings contain more character details and are easier to read when compared with pictures of the original steles.

In this article, we consider a research dataset of 20 stele images[2]. It encompasses all steles, for which we have obtained ground truth information so far at the level of individual characters, i.e. bounding boxes around the characters as well as their machine-readable Chu Nom transcription in unicode. Characters that are not readable are marked with a special symbol. In total, the dataset contains 5,138 characters, which corresponds to an average of about 257 characters per stele.

4 Hybrid deep learning

Figure 2 provides an overview of the proposed hybrid deep learning method for keyword spotting. At the core of the method is a deep learning model that is responsible to detect the location of main text characters on the stele images. It is trained on synthetic data and auto-calibrated to real data. Afterwards, a structural representation and comparison with keyword templates is performed for spotting.

The training data used for supervised learning of the deep learning model does not originate from human annotations. Instead, human knowledge is used to design the synthetic training data and to perform the auto-calibration. Also, human knowledge is used to model the characters with a graph-based representation and to perform the structural matching with the keyword templates based on heuristic methods.

In the following, the individual components are described in more detail.

4.1 Character detection

The deep learning model is an object detection network with a you look only once (YOLO) [20] architecture, which has originally been introduced for detecting objects in natural scenes, e.g. pedestrians in the context of autonomous driving. When applied to the problem of character detection on stele images, one of the main differences is that a large number of small objects need to be detected, rather than a small number of large objects. Therefore, it is important that the visual analysis is performed with a sufficiently high resolution, such that even small strokes of the logographic characters can be taken into account.

Two initial preprocessing steps are applied to the original images:

- Rescaling the stele images to a uniform height of 1024 pixels, while keeping the same aspect ratio.
- Inverting the colors, such that the engraved characters appear in dark color rather than white.

The height has been chosen to ensure that the stele images fit into the GPU memory. The resulting images are typically smaller than the originals. Inverting the colors has been chosen with respect to improved readability and more convenient generation of synthetic data.

The specific network architecture used for character detection is YOLOv5 [21], which analyzes the image at multiple sufficiently large scales to detect small characters on large steles. The *backbone* of YOLOv5 is a cross stage partial network (CSPNet) [22], which extracts convolutional feature maps. The *neck* is a path aggregation network (PANet) [23], which performs a combination of feature maps at different scales. Finally, a dual *head* is used to perform both classification and bounding box regression on the combined feature maps. Theoretically, the dual head would allow us not only to localize the characters but also

[2] The dataset is available at `https://github.com/asciusb/steles_kws_database`.

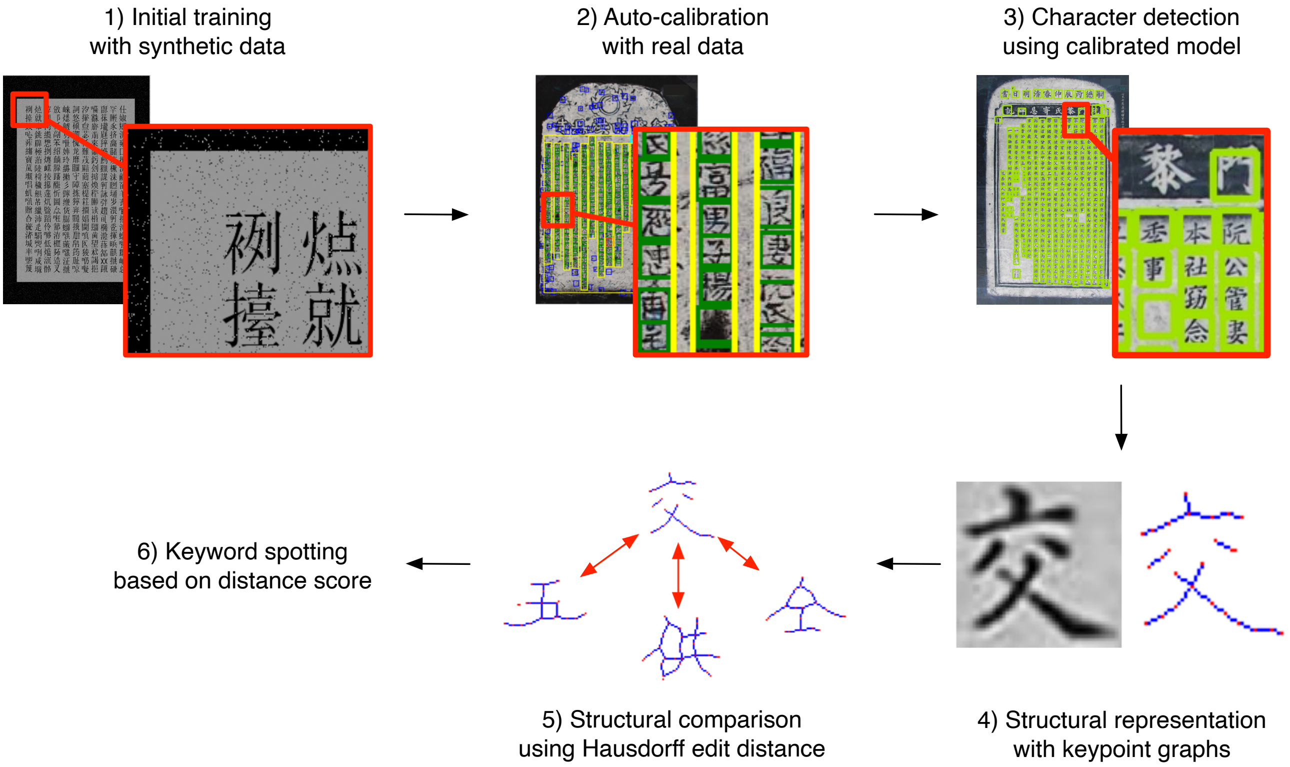


Figure 2: Overview of the hybrid deep learning method for keyword spotting. Green boxes represent detected characters and yellow boxes are detected columns.

to classify them. However, practical attempts to classification have failed when considering thousands of different Chu Nom character classes. Instead, the classification head only performs a binary classification, whether or not a character is present, and the regression head predicts the extent of the character bounding box.

4.2 Synthetic training

The initial training of the character detection network is based on thousands of fully synthetic training steles (Figure 2, step 1), for which the ground truth annotations, i.e. bounding boxes around the characters, is generated alongside with the synthetic images. The generation is guided by the following heuristic considerations:

- Color-inverted stele images contain dark character engravings on a gray stone background, surrounded by a black border.
- Characters are arranged in a column layout.

The data generation therefore proceeds as follows. First, a gray rectangle is drawn on a black background. Then, a Chu Nom font[3] is used to write random text on the gray rectangle in a random number of columns. Finally, random noise is added to the synthetic images by means of translation, blur, changes in brightness, as well as salt and pepper noise, in order to avoid overfitting of the character detection network.

4.3 Auto-calibration

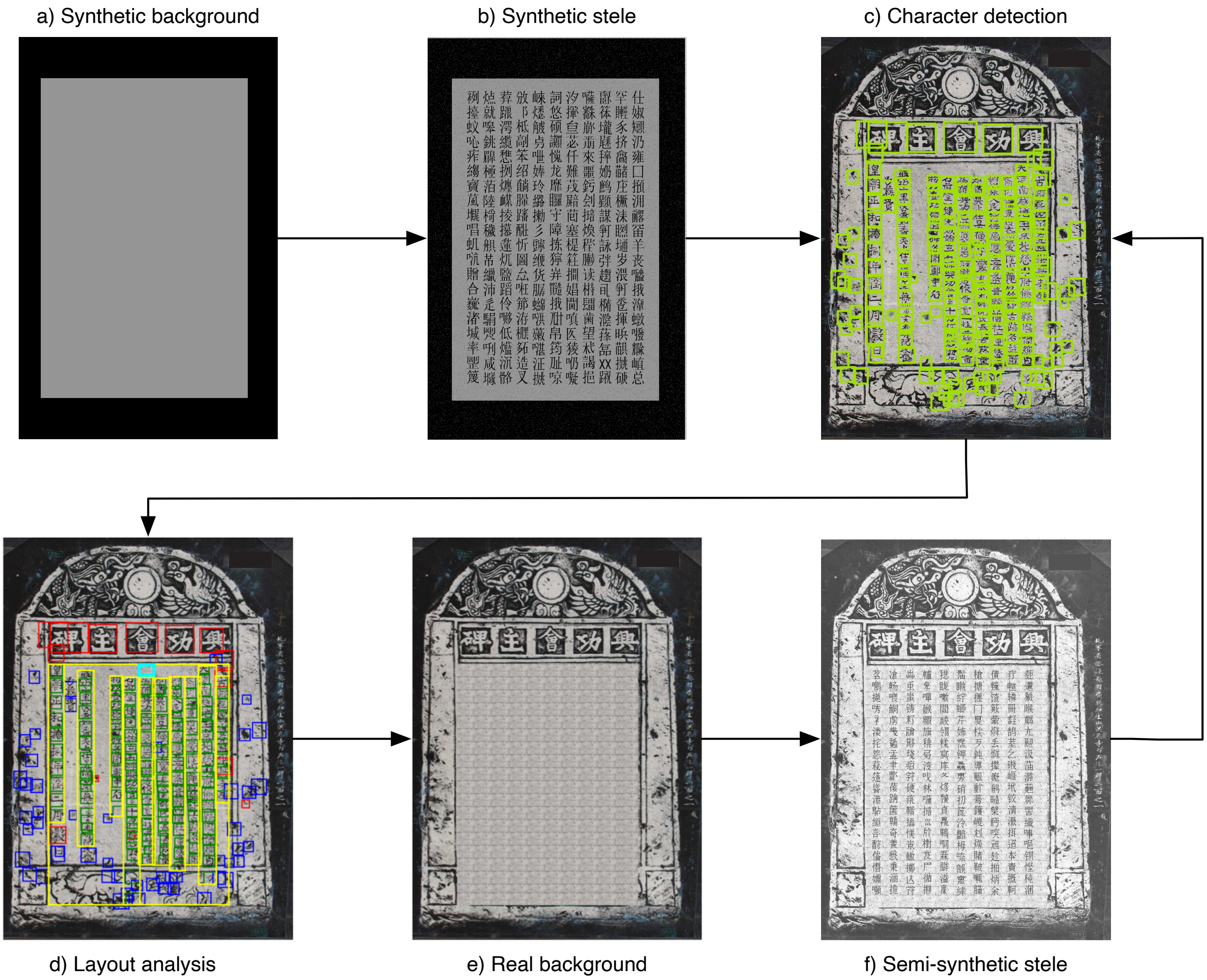
After an initial training on synthetic data, the network is applied and adapted to real data by means of auto-calibration (Figure 2, step 2), following the procedure introduced in [13]. The aim of the auto-calibration is to replace the generic gray rectangle and black border of the synthetic training data with real stele backgrounds, such that the network can improve the separation between stele background and character foreground.

The following heuristic considerations are taken into account for detecting the main text area:

- Main text characters have approximately the same size and are organized in columns.
- The main text area is rectangular.

The auto-calibration is illustrated in more detail in Figure 3. After printing random Chu Nom text on simple backgrounds to create fully synthetic stele images, an initial training of the character detection network is performed. Afterwards, the network is applied to real stele images and layout analysis is used to recognize the main text area. Layout analysis consists of the following steps. The median

[3] The NomNaTongLight font available at http://www.nomfoundation.org.



a) Synthetic background

b) Synthetic stele

c) Character detection

d) Layout analysis

e) Real background

f) Semi-synthetic stele

Figure 3: Auto-calibration of the deep learning model for character detection. Green boxes represent detected characters, red and blue boxes are characters discarded during layout analysis, yellow boxes are columns and the main text area, and the cyan box is a homogeneous background region.

box is calculated to estimate the size of the main text characters. Characters that are either too small (e.g. parts of ornaments or parts of the background), or too large (e.g. characters of the title above the main text) are discarded. Afterwards, unsupervised clustering using the DBSCAN [24] algorithm is performed to find the main text columns and thus the main text area around the columns (yellow rectangles in Figure 3). A homogeneous non-text region with low variance is determined as a pattern to fill the entire main text area (cyan rectangle in Figure 3). Finally, the Chu Nom font is used to write synthetic printed text on the main text area, similar to the generation of the initial training data, with the difference that a real stele background is present around the printed Chu Nom text.

The auto-calibration leads to new semi-synthetic training data, on which the initial network is further fine-tuned, thus adapting to real stele backgrounds and improving the detection accuracy. For further details, we refer to [13].

4.4 Structural representation

Once the characters have been detected by the calibrated network (Figure 2, step 3), the character images are modeled with a graph-based representation that captures their structure, i.e. the arrangement of individual strokes that constitute the character (Figure 2, step 4). We employ *keypoint graphs* [25], which have been used successfully for keyword spotting in the past for Latin manuscripts [6] as well as Chu Nom manuscripts [16] written with ink on parchment or paper.

The graph extraction is illustrated in Figure 4. First, a local text enhancement is applied by means of a difference of Gaussians (DoG) filter. Afterwards, the image is binarized with a global threshold and thinned to obtain strokes that have a width of one pixel. Endpoints and intersection points are added as nodes to the keypoint graph, labeled with their (x, y) coordinates. For circular structures, a random point is

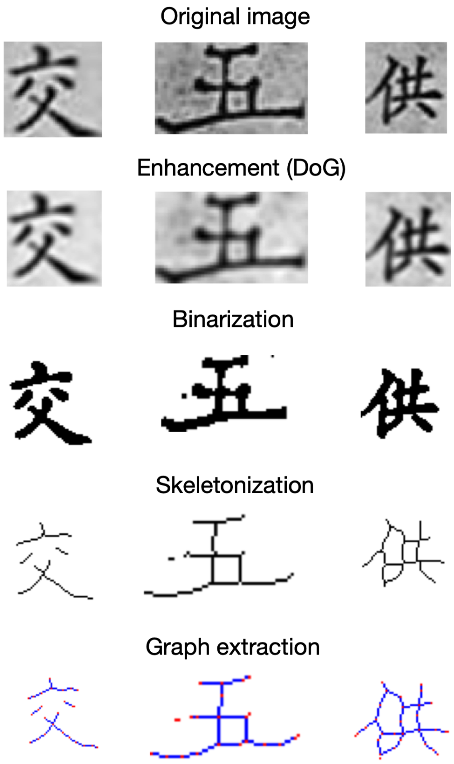


Figure 4: Keypoint graph extraction.

added as a node as well. To complete the initial set of nodes, additional points are added as nodes at regular intervals of D pixels on the skeleton image. Once all nodes have been added to the graph, their coordinate labels are normalized to zero mean and unit variance (z-score). Finally, neighboring nodes on the skeleton image are connected with unlabeled and undirected edges. For more details on the graph extraction, we refer to [26], whose implementation of keypoint graphs is used in the present article.

An important modification of the keypoint graph extraction, which led to successful spotting results on the stele images was to model the characters in super-resolution, in order to capture sufficient details about small strokes that may mark the difference between two similar Chu Nom engravings. To that end, the bounding box of the detection network is first translated back to the original image (inverting the downscaling to 1024 pixel height), then the character is cut out from the original image and upscaled to the same width S for all characters, while keeping the aspect ratio. When extracting graphs from a character image in super-resolution, i.e. when using values of S larger than the original width, it is possible that strokes in the keypoint graph contain more nodes than pixels in the original image. Hence, even very small strokes become more relevant in the graph-based representation.

4.5 Structural comparison

To compare the graphs of the character images with keyword graphs (Figure 2, step 5), we consider the graph edit distance [27, 28]. It is a general graph dissimilarity measure that is applicable to any kind of labeled graphs. With respect to a set of basic edit operations, typically insertion, deletion, and label substitution for nodes and edges, it calculates the minimum edit cost for transforming one graph into another. However, the exact graph edit distance is more of theoretical value than of practical relevance because it is an NP-complete problem, which can only be solved for small graphs with few nodes in reasonable time.

In order to cope with large character graphs, which may have over 100 nodes in super-resolution, we use the Hausdorff edit distance [17], an approximation of graph edit distance that calculates a lower bound in quadratic time. Derived from the Hausdorff distance between sets, it compares each node $u \in g_1$, plus its adjacent edges, of one graph g_1 with every node $v \in g_2$, plus its adjacent edges, of another graph g_2 and sums up the minimum edit cost $f(u, v)$ for matching the substructures. A special ϵ node is considered for insertions (ϵ, v) and deletions (u, ϵ). Formally,

$$\begin{aligned} HED(g_1, g_2) &= \sum_{u \in g_1} \min_{v \in g_2 \cup \{\epsilon\}} f(u, v) \\ &\quad + \sum_{v \in g_2} \min_{u \in g_1 \cup \{\epsilon\}} f(u, v) \end{aligned}$$

We use the Euclidean cost model for the structural comparison of keypoint graphs. It considers the Euclidean distance of the (x, y) labels for node label substitution, a constant cost c_n for node insertion and deletion, and a constant cost c_e for edge insertion and deletion.

4.6 Keyword spotting

To spot a Chu Nom character (Figure 2, step 6), n template images of the keyword are collected from real steles and keypoint graphs are extracted. Afterwards, the minimum HED score

$$score(g) = \min_{t \in T} HED(g, t)$$

is calculated for each character graph g of the stele images with respect to the template graphs $T = \{t_1, \ldots, t_n\}$. Finally, the character graphs are sorted according to the spotting score, such that the most similar character graphs appear in the top ranks.

For evaluating the spotting performance, precision (P) and recall (R) are calculated as

$$\begin{aligned} P &= \frac{TP}{TP + FP}, \\ R &= \frac{TP}{TP + FN}, \end{aligned}$$

with respect to the number of true positives (TP), false positives (FP), and false negatives (FN) for each possible score threshold. Then, the average precision (AP) is calculated for each keyword and the mean average precision (mAP)

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i$$

over all K keywords is used as the final performance measure for keyword spotting.

5 Experiments

5.1 Spotting scenarios

To evaluate the proposed hybrid deep learning method, the 5,138 ground truth characters of the 20 stele images (see Section 3) are randomly separated into three distinct sets for template selection (50%), validation (25%), and testing (25%), respectively.

The template selection set is used to select $n = 5$ templates per keyword, the validation set is used for optimizing hyper-parameters, and the test set is used for evaluating the final spotting performance. A total of $K = 128$ keywords are spotted, which appear at least 5 times in the template selection set, at least once in the validation set, and at least once in the test set.

We compare three spotting scenarios with respect to the use of human annotations, as listed in Table 1:

- The **fully annotated** scenario uses ground truth labels for parameter optimization as well as performance evaluation.
- The **font-validated** scenario does not require human annotations for parameter optimization. Instead, a synthetic font-based validation set is used (see below).
- The **annotation-free** scenario, which is the target scenario for the proposed method, does not require any human annotations. It evaluates the keyword spotting system with respect to automatically detected characters instead of ideal ground truth locations.

The synthetic font-based validation set is created as follows. 20 keywords are selected randomly and printed in 5 different Chu Nom fonts. 900 other characters are printed and added to the validation set, which is composed of 1,000 characters in total. Each of the keywords is then spotted on the validation set using a single template and the mAP results are used to compare and optimize different parameter settings.

Furthermore, we compare two spotting scenarios with respect to the engraving styles:

- The **same style** scenario spots keywords on each stele image separately, such that the engraving style of the keyword templates is the same as the style of the stele characters.
- The **mixed styles** scenario spots keywords across all 20 stele images, taking into account different engraving styles.

5.2 Parameter optimization

For the character detection network (see Sections 4.2 and 4.3), we consider only one set of hyper-parameters, i.e. the default parameters of the medium-sized YOLOv5m model[4]. The weights of the model are pretrained on the COCO [29] object detection dataset. The pretrained network is fine-tuned with 30,000 synthetic steles over 15 epochs until convergence. Afterwards, an additional fine-tuning epoch is used for auto-calibration with real stele backgrounds.

The parameters of the structural representation and the structural comparison (see Sections 4.4 and 4.5) are optimized in two steps. First, a default character width of $S = 150$ pixels and node distance of $D = 5$ pixels is fixed to evaluate a range of node and edge costs $c_n, c_e \in \{0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1\}$ on the validation set. Afterwards, the optimal node and edge costs are fixed and different character widths $S \in \{90, 120, 150, 180, 210\}$ and node distances $D \in \{3, 4, 5, 6, 7\}$ are evaluated on the validation set.

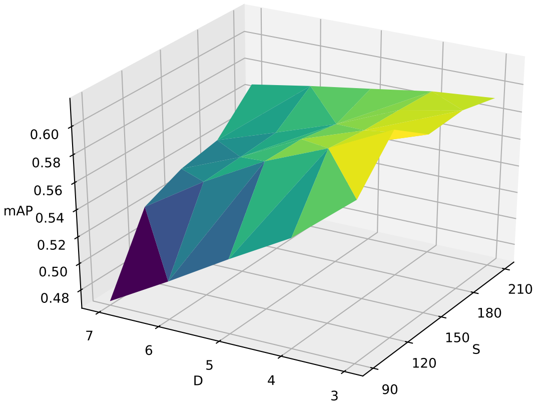


Figure 5: Optimization of structural representation for the fully annotated scenario.

The optimal parameters are listed in Table 2 for both the fully annotated and the font-validated scenario. When considering synthetic printed characters (font-validated), we observe that a larger node distance and a larger edge cost is preferred when compared with real stele characters (fully annotated). This may be due to the increased stability, i.e. less variability, of the character shapes and character background in the case of printed fonts.

A more detailed view on the optimization of S and D is provided in the three-dimensional visualizations in Figures 5 and 6. They show that changing the parameters lead to more significant differences in mAP for the real characters when compared with synthetic ones, indicating that the synthetic validation set may need to be improved to better represent the challenges encountered for real characters.

[4] github.com/ultralytics/yolov5, commit cc03c1d5727e178438e9f0ce0450fa6bdbbe1ea7

Table 1: Different keyword spotting scenarios with respect to human annotations.

	Parameter Optimization	**Performance Evaluation**
Fully annotated	Ground truth validation set	Ground truth test set
Font-validated	Synthetic font-based validation set	Ground truth test set
Annotation-free	Synthetic font-based validation set	Automatic detection test set

Table 2: Optimal parameters for structural representation and comparison.

Parameters	**Fully annotated**	**Font-validated**
Character width S	120	150
Node distance D	3	7
Node cost c_n	0.9	0.3
Edge cost c_e	0.9	2.1

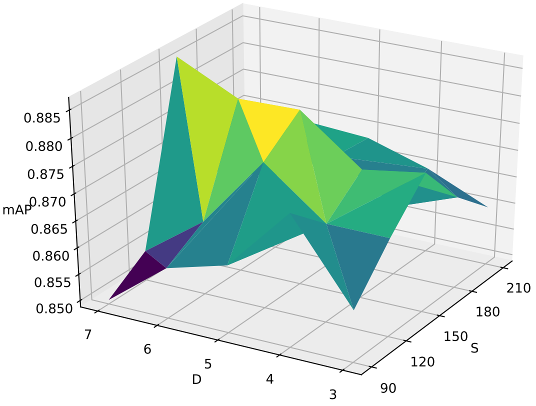
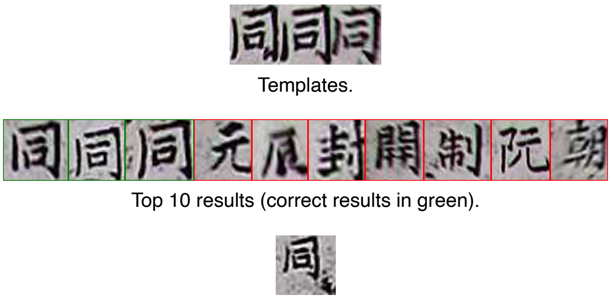


Figure 6: Optimization of structural representation for the font-validated scenario.

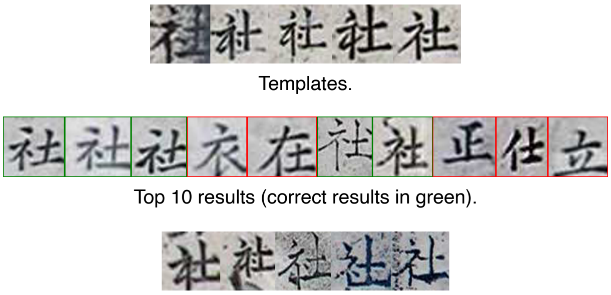


Templates.

Top 10 results (correct results in green).

Keywords existing in the test set, but not found in the top 10 results.

Figure 7: Qualitative spotting results for the same style spotting scenario.



Templates.

Top 10 results (correct results in green).

Keywords existing in the test set, but not found in the top 10 results.

Figure 8: Qualitative spotting results for the mixed styles spotting scenario.

5.3 Runtime performance

For training the YOLO-based character detection model, we used 2 Nvidia Titan RTX GPUs. One training epoch on 30,000 stele images took 6.3 minutes on average and a total of 16 training epochs was sufficient for convergence. Detecting characters with the trained YOLO network took only a few milliseconds per stele.

For graph matching with the Hausdorff edit distance, we used computational nodes with 64 CPU cores (AMD EPYC, 2.25GHz). One graph comparison took 4.4 milliseconds on average, which allowed us to spot a single keyword template in about 1 second per stele.

Note that the graph comparisons need to be performed only once. Afterwards, the positions of all keywords in the collection of stele images can be indexed, such that historians can search and retrieve keywords quasi instantly based on the index.

5.4 Spotting performance

Table 3 shows the spotting performance on the test set when using optimized parameters. The results obtained for the steles are put into context with results obtained for Chu Nom manuscripts reported in [16]. These manuscripts are written with ink on parchment with a regular writing style, leading to better character detection quality and less noise in the character images when compared with the steles. The results are not directly comparable, because a different set of keywords was used, but they provide a point of reference for a less challenging keyword spotting task.

For the same style spotting scenario, the performance level is excellent with a mAP of 72% for the annotation-free scenario. The performance is similar to that of manuscripts, which typically are better readable and have less varia-

Table 3: Spotting performance in terms of mean average precision (mAP) on the test set.

	Fully annotated	**Font-validated**	**Annotation-Free**
Manuscripts (Kieu) [16]	0.76	0.78	0.77
Steles: Same style	0.85	0.81	0.72
Steles: Mixed styles	0.56	0.50	0.40

tions of the writing style when compared with stele images. However, the impact of optimizing the parameters on synthetic characters, rather than real ones, is stronger (mAP reduced from 85% to 81%) and the loss in mAP for automatic character detection is stronger as well (mAP reduced from 81% to 72%). These results show the increased difficulty of spotting Chu Nom characters in stele images and leave room for improvements regarding parameter optimization and character detection.

For the mixed styles spotting scenario, the performance drops significantly to a mAP of 40% for the annotation-free scenario. It indicates a limitation of the proposed hybrid deep learning method, which did not generalize well to engraving styles that are different from the keyword templates. An application to stele collections with similar engraving styles seems therefore more promising. Note, however, that we have only used 5,138 characters in our experiments. It is possible that the method will better generalize with a larger dataset.

Figures 7 and 8 provide qualitative spotting results for both scenarios. For same style spotting, 3 out of 4 characters are correctly spotted in the first 3 ranks. However, the fourth character is not part of the top 10 results, because of an error in automatic character detection, which has included some noise in the bottom right corner. For mixed style spotting, 5 out of 10 characters appear in the top 10 ranks, but not in the first 5 ranks. The remaining 5 characters are not part of the top 10 results, due to noise but also due to different engraving styles, which are not represented in the keyword template images.

6 Conclusions

The proposed hybrid deep learning approach to keyword spotting aims to combine the strengths of data-driven methods with knowledge-based modeling. In a first step, a deep convolutional neural network is trained on a large synthetic dataset to detect printed Chu Nom characters. By means of self-calibration, the network is then automatically adapted to the stele images. In a second step, the detected characters are modeled by means of keypoint graphs and the Hausdorff edit distance is used to efficiently perform a structural comparison for retrieving keywords.

Especially when the engraving style of the keyword is the same as the style of the stele characters, an excellent mean average performance of over 70% is achieved. In the case of mixed engraving styles, however, the spotting results drop to about 40% mean average precision. Although this performance level is still helpful for historians to browse large image collections of heterogeneous steles, there is clearly room for improvement.

There are several interesting lines of future research to further improve the results. Staying in the same style scenario, future work includes the investigation of style clustering, such that similar engraving styles can be identified across a large number of stele images. Noise removal methods may also be interesting to avoid spotting mistakes due to non-character artifacts.

With respect to the mixed style scenario, it may be necessary to perform some sort of data-driven learning to improve the spotting results, for example by means of geometric deep learning with graph neural networks [30]. In order to avoid the requirement of human annotations, it would be interesting to pursue a self-calibration strategy similar to the self-calibration of the character detection network.

Finally, a promising line of future research would be to generalize the proposed spotting method to other historical scripts and languages.

References

[1] Andreas Fischer, Marcus Liwicki, and Rolf Ingold, editors. *Handwritten Historical Document Analysis, Recognition, and Retrieval — State of the Art and Future Trends*. World Scientific, 2020.

[2] Raghavan Manmatha, C. Han, and E.M. Riseman. Word spotting: A new approach to indexing handwriting. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pages 631—637, 1996.

[3] T. M. Rath and R. Manmatha. Word spotting for historical documents. *Int. Journal on Document Analysis and Recognition*, 9:139–152, 2007.

[4] K. Terasawa and Y. Tanaka. Slit style HOG features for document image word spotting. In *Proc. 10th Int. Conf. on Document Analysis and Recognition*, pages 116–120, 2009.

[5] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Proc. 11th Int. Conf. on Document Analysis and Recognition*, pages 63–67, 2011.

[6] Michael Stauffer, Andreas Fischer, and Kaspar Riesen. *Graph-based Keyword Spotting*. World Scientific, 2019.

[7] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character HMMs. *Pattern Recognition Letters*, 33(7):934–942, 2012.

[8] L. Rothacker, M. Rusiñol, and G. A. Fink. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In *Proc. 12th Int. Conf. on Document Analysis and Recognition*, pages 1305–1309, 2013.

[9] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(2):211–224, 2012.

[10] Sebastian Sudholt and Gernot A Fink. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 277–282. IEEE, 2016.

[11] Thai V. Hoang, Salvatore Tabbone, and Ngoc-Yen Pham. Extraction of nom text regions from stele images using area voronoi diagram. In *10th International Conference on Document Analysis and Recognition*, pages 921–925, 2009.

[12] Anna Scius-Bertrand, Lars Voegtlin, Michele Alberti, Andreas Fischer, and Marc Bui. Layout analysis and text column segmentation for historical vietnamese steles. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, pages 84–89, 2019.

[13] Anna Scius-Bertrand, Michael Jungo, Beat Wolf, Andreas Fischer, and Marc Bui. Annotation-free character detection in historical Vietnamese stele images. In *Proc. 16th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 432–447, 2021.

[14] Jonas Diesbach, Andreas Fischer, Marc Bui, and Anna Scius-Bertrand. Generating synthetic styled chu nom characters. In *Proc. 18th Int. Conf on Frontiers in Handwriting Recognition (ICFHR)*, 2022.

[15] Kha Cong Nguyen, Cuong Tuan Nguyen, and Masaki Nakagawa. Nom document digitalization by deep convolution neural networks. *Pattern Recognition Letters*, 133:8–16, 2020.

[16] Anna Scius-Bertrand, Linda Studer, Andreas Fischer, and Marc Bui. Annotation-free keyword spotting in historical vietnamese manuscripts using graph matching. In *IAPR Joint International Workshops on Statistical Techniques in Pattern Recognition (SPR 2022) and Structural and Syntactic Pattern Recognition (SSPR 2022) : S+SSPR*, 2022.

[17] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke. Approximation of graph edit distance based on Hausdorff matching. *Pat. Rec.*, 48(2):331–343, 2015.

[18] A. Scius-Bertrand, A. Fischer, and M. Bui. Retrieving keywords in historical vietnamese stele images without human annotations. In *Proc. 11th Int. Symposium on Information and Communication Technology (SoICT)*, 2022.

[19] Philippe Papin. Aperçu sur le programme "publication de l'inventaire et du corpus complet des inscriptions sur stèles du viêt-nam". *Bulletin de l'École française d'Extrême-Orient*, 90(1):465–472, 2003.

[20] 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*, pages 779–788, 2016.

[21] Glenn et al. Jocher. ultralytics/yolov5: v4.0 - nn.silu() activations, weights & biases logging, pytorch hub integration. DOI: 10.5281/zenodo.4418161, 2021.

[22] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[23] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[24] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[25] A. Fischer, K. Riesen, and H. Bunke. Graph similarity features for HMM-based handwriting recognition in historical documents. In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, pages 253–258, 2010.

[26] Paul Maergner, Vinaychandran Pondenkandath, Michele Alberti, Marcus Liwicki, Kaspar Riesen, Rolf Ingold, and Andreas Fischer. Combining graph edit distance and triplet networks for offline signature verification. *Pattern Recognition Letters*, 125:527–533, 2019.

[27] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[28] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Trans. on Systems, Man, and Cybernetics*, 13(3):353–363, 1983.

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[30] Pau Riba, Andreas Fischer, Josep Lladós, and Alicia Fornés. Learning graph edit distance by graph neural networks. *Pattern Recognition*, 120:108132, 2021.