Retrieving Keywords in Historical Vietnamese Stele Images Without Human Annotations

Anna Scius-Bertrand
iCoSys, HES-SO
Fribourg, Switzerland
DIVA, University of Fribourg
Fribourg, Switzerland
anna.scius-bertrand@hefr.ch

Andreas Fischer
iCoSys, HES-SO
Fribourg, Switzerland
DIVA, University of Fribourg
Fribourg, Switzerland
andreas.fischer@hefr.ch

Marc Bui
Ecole Pratique des Hautes Etudes
Paris, France
marc.bui@ephe.psl.eu

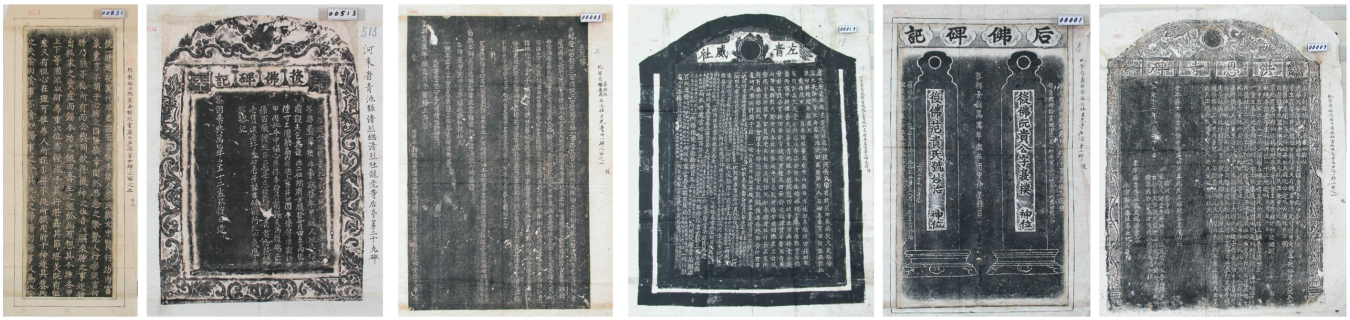

Figure 1: Example stele images.

ABSTRACT

Stone engravings on Vietnamese steles are an invaluable resource for historians to study the life of the villagers in the past. Thanks to pictures taken of stampings of the steles, they can be investigated today in the form of digital images. Automatic keyword spotting is a promising means to access the textual content of the images, allowing to retrieve steles that contain a certain query term. In this paper, we present a complete pipeline for retrieving Chu Nom characters in Vietnamese steles that operates fully automatically on the original images, without the need for preprocessing, segmentation, or human annotation. It combines a self-calibration approach to character detection using deep convolutional neural networks with a graph-based approach to keyword spotting that compares templates of the search term with detected characters based on structural properties.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**; **Visual content-based indexing and retrieval**.

KEYWORDS

keyword spotting, historical documents, Vietnamese stele images, Chu Nom, annotation-free, Hausdorff edit distance

SoICT 2022, December 1–3, 2022, Hanoi, Vietnam

ACM ISBN 978-1-4503-9725-4/22/12.
https://doi.org/10.1145/3568562.3568606

ACM Reference Format:
Anna Scius-Bertrand, Andreas Fischer, and Marc Bui. 2022. Retrieving Keywords in Historical Vietnamese Stele Images Without Human Annotations. In *The 11th International Symposium on Information and Communication Technology (SoICT 2022), December 1–3, 2022, Hanoi, Vietnam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3568562.3568606

1 INTRODUCTION

Stone engravings on large steles found in Vietnamese villages are an important resource for historians. They allow to study the history of Vietnam from the perspective of the villagers and are thus complementary to historical manuscripts, which are focused on the royal court and the clergy. In the Vietnamica project[1], the goal of the historians is to study the steles by means of a large body of digital images, which were acquired by taking pictures of stampings.

To access the contents of the images, automatic document analysis methods could greatly support the efforts of the historians. However, despite strong progress in the past decades, historical document analysis remains a challenging problem and an active field of research [2]. As illustrated in Figure 1, the steles contain a large variability of layouts, ornamentation, engraving styles, and artifacts due to damages. To cope with such variation, the current state of the art in document analysis is mostly based on deep convolutional neural networks [8, 17], which are trained with human-annotated learning samples.

Unfortunately, comprehensive annotations do not yet exist for the steles and are difficult to obtain: first, because a large number of steles would be required to cover the different layouts and engraving styles and, secondly, because expert knowledge is needed to read the ancient Chu Nom script.

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

Nevertheless, initial attempts have been made to automatically analyze the stele images, including a rule-based approach to layout analysis using Voronoi diagrams [5], a learning-based approach to layout analysis using convolutional U-Nets with only few approximate annotations [15], and an annotation-free method to character detection [13], which uses self-calibration for convolutional object detection networks to perform transfer learning from synthetic pages with printed characters to real stele images.

In this paper, we go a step further and aim to automatically identify keywords on the stele images. For this purpose, we propose a complete pipeline that does not require any preprocessing of the images or human-annotated learning samples to retrieve search terms. That is, it can directly be applied to the raw images and can thus be useful for an initial analysis of the document collection by the historians.

The pipeline consists of two components: First, we extract individual character images by means of self-calibrated object detection networks as proposed in [13]. Afterwards, we use a graph-based representation of the character images as well as the query term, to match the structure of the characters and retrieve the most similar character instances. The graph-based approach has already been successfully applied to historical Latin manuscripts [16] and, more recently, also to historical Chu Nom manuscripts [14]. We adapt the method to the stone engravings and perform an experimental evaluation of the entire annotation-free pipeline, to investigate its potential and limitations for keyword spotting in Vietnamese stele images.

The remainder of the paper is organized as follows. First, the dataset is introduced in Section 2, followed by a description of the keyword spotting system in Section 3 and the experimental evaluation in Section 4. Finally, we draw some conclusions in Section 5.

2 DATASET OF VIETNAMESE STELE IMAGES

Our dataset is issued from a collection of inscriptions on steles realized by the Ecole Française d'Extrême Orient (EFEO) and Institut Han-Nôm. The inscriptions on steles have been preserved by means of stamping. This is a technique that consists of reproducing the content of a stele on a sheet of paper. To do this, a binder, for example banana juice, is put on the stone where a sheet of paper is attached. Then, the stamper applies a roller coated with ink on the entire surface of the stele. Thus, all the raised parts of the stele will appear in black and the hollows in the stele, such as the inscriptions, will appear in white.

A study by P. Papin [9] gives us information on the dating, the places and the historical interest of the steles. Most of the steles were erected between the XVII^{me} and the XVIII^{me} century, i.e. in the middle of the restored Le dynasty. This period corresponds to a period of weakening of the central power leaving more freedom to the countryside. Thanks to this gain in autonomy, they developed and recorded the changes in the daily life of the villages on steles.

The steles come mainly from the north of Vietnam (about 80% of the corpus) and more precisely from the five provinces around Hanoi which represent half of the corpus, and a third of the corpus comes from the three provinces: Ha-Dong, Hai-Duong and Vinh-Yên. It should be noted that these locations not concern the south of the country. In the north, they specifically concern villages that are rich enough to be able to build steles. That is to say, where they were able to develop resources and trade, mainly in the rice-growing plains and near waterways.

The vast majority of the corpus is on the scale of a village or even a neighborhood of a village. These are internal documents of the villages, sometimes of a peasant family itself. The majority of the stele are religious steles of donations. Some steles are copies on stone of minutes written on paper. This allowed decisions that had been made to be displayed for all to see, without fear of their deterioration due to weather conditions or armed conflicts. The steles can also deal with questions of demarcation, markets, finances or public construction. They are rich in social, economic, religious and linguistic information about the villages.

In this paper we used a dataset[2] of all the stele images for which we had a transcription, with a total of 8 steles including 2,786 characters. The ground truth used for quantitative evaluation of our keyword spotting system contains the exact character location (bounding box) of each character on the stele images, together with its Chu Nom unicode character.

3 KEYWORD SPOTTING SYSTEM

The proposed pipeline for annotation-free keyword spotting has two components. First, a character detection network is trained on synthetic page images and an unsupervised self-calibration to the stele images is performed. Afterwards, a graph-based representation is extracted from the detected character images as well as from several template images of the search term. Finally, graph matching is performed to retrieve the most similar characters. The two system components are detailed below, followed by a description of the evaluation measure used to evaluate the spotting performance.

3.1 Character Detection

For automatic character detection, we rely on a YOLO network [10] and the self-calibration method introduced in [13]. In contrast to typical object detection tasks, which aim to detect one or few objects in a natural scene, the goal of character detection is to find a large number of small characters within a large document image. The architecture of YOLOv5 used in our system is well-suited for this purpose, as it performs detection at several scales, including large scale with high resolution. It uses a Cross Stage Partial Network (CSPNet) [18] as a backbone to extract convolutional feature maps from the image. Afterwards, a Path Aggregation Network (PANet) [6] is used as a neck to combine the feature maps across different scales. Finally, a dual head performs both character classification as well as bounding box regression based on the feature maps. For classification, a binary decision for being a character or not is considered, instead of attempting to classify each Chu Nom character individually.

The self-calibration process is illustrated in Figure 2. It starts by generating synthetic stele images with a black border and a gray background, on which printed Chu Nom characters are inserted based on a font. Variations of the stele images are generated by means of translation, salt and pepper noise, blur, and brightness changes. After training an initial network on the synthetic data, it is applied to the real stele images and the detection results are

[2]Available here: https://github.com/asciusb/steles_kws_database

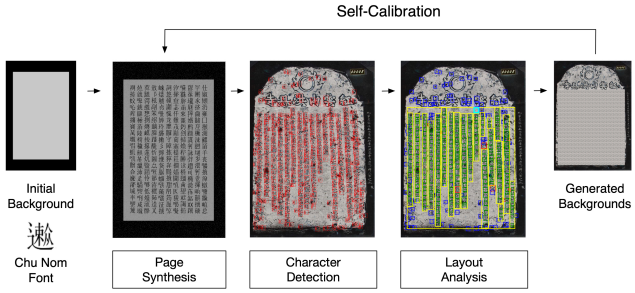

Figure 2: Self-calibration. Figure from [13]

interpreted by means of text column clustering, in order to estimate the main text area. The main text area is then filled with a homogeneous low-variance region of the stone background and printed Chu Nom characters are inserted again. The resulting stele images are still synthetic but incorporate real stele images around the main text area. To calibrate the character detection network to the steles, it is retrained on this more realistic training set. For further details, we refer the reader to [13].

3.2 Graph Matching

For graph-based keyword spotting, we extract keypoint graphs [3] from the stone engravings that are matched by means of the Hausdorff edit distance [4], similar to previous work on keyword spotting in Latin manuscripts [16] and historical Vietnamese manuscripts [14] written with ink on parchment or paper.

Figure 3: Graph extraction.

A labeled graph $g = (V, E, \mu, \nu)$ is a finite set of nodes V that are linked with edges $E \subseteq V \times V$. The functions $\mu : V \rightarrow L_V$ and $\nu : E \rightarrow L_E$ can be used to assign labels to the nodes and edges, respectively. Labels can be of any type, including symbols, weights, or vectors.

Figure 3 illustrates the extraction of keypoint graphs. First, the detected character image is transformed to grayscale and the edges are enhanced by means of a Difference of Gaussian (DoG) filter. Then, the image is binarized with a global threshold and the stroke is reduced to one pixel width, in order to obtain a skeleton of the handwriting. The following keypoints are identified: endpoints, junctions, and additional points sampled at distance D in between. They constitute the set of nodes V of a keypoint graph, labeled with their (x, y) coordinates, and are linked with an unlabeled edge if they are neighbors on the skeleton. In this paper, we use the software implementation of [7] to extract the keypoint graphs.

The same graph extraction is applied to template images of the search term that are manually selected by the human user (query by example), or to printed images of the search term (query by string).

An important adaptation for keyword spotting in Vietnamese steles was to rely on a super-resolution of the characters. To normalize different resolutions across different stele images, all Chu Nom characters are rescaled to the same width S, keeping the aspect ratio. By choosing S larger than the original character width, super-resolution is achieved, allowing the graph to contain more nodes for a stroke than stroke pixels in the original image. By means of this upscaling, it is possible to highlight even small strokes that are often important features to distinguish Chu Nom characters.

After graph extraction, the concept of graph edit distance [1, 12] is used to perform a structural matching between the search term and the document graphs. This distance measure is based on the idea of transforming one graph into the other by means of edit operations, including label substitution, as well as deletion and insertion of nodes and edges. By assigning costs to the edit operations, domain knowledge is integrated, such that significant changes of the graph result in large costs. For the keypoint graphs, we employ an Euclidean cost function, which assigns the Euclidean distance for label substitution and a constant cost c_V, c_E for node and edge deletion and insertion, respectively. The graph edit distance is then defined as the minimum cost for transforming one graph to the other.

Unfortunately, the exact computation of graph edit distance is NP-complete, which makes it unfeasible for the Chu Nom graphs, which may contain over 100 nodes in super-resolution. Instead, we rely on a recently introduced approximation, the Hausdorff Edit Distance (HED) [4], which computes a lower bound in quadratic time:

$$HED(g_1, g_2) = \sum_{u \in V_1} \min_{v \in V_2 \cup \{\epsilon\}} f(u, v) + \sum_{v \in V_2} \min_{u \in V_1 \cup \{\epsilon\}} f(u, v) \quad (1)$$

where $f(u, v)$ computes the cost of matching the local substructure around a node in the first graph $u \in V_1$ and its adjacent edges with the local substructure of a node in the second graph $v \in V_2$ and its adjacent edges. The special empty node ϵ refers to deletions and insertions. For more details on the function $f(u, v)$ we refer to [4].

To compute the keyword spotting score for character graph g with respect to $\mathcal{T} = \{t_1, \ldots, t_n\}$ template graphs of the search term, we consider the minimum HED:

$$score(g) = \min_{t \in \mathcal{T}} HED(g, t) \quad (2)$$

3.3 Performance measure

To evaluate the performance of keyword spotting, the average precision (AP) is computed for each query keyword individually over all possible recall values, i.e. over all possible score thresholds. Then, the mean average precision (mAP) with respect to all N

keywords is used as a global evaluation measure:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

4 EXPERIMENTAL EVALUATION

To evaluate the performance of our method, we first describe the hyper-parameter of our system and their optimization without using human annotations. Then we detail the final test setup for keyword spotting and present the results.

4.1 Parameters optimization

To perform the optimization of the hyper-parameters, we have separated our database (see Section 2) in the following way: 50% of the characters have been used to select the keywords to be searched, 25% of the characters are reserved for validation and 25% are considered as test set to evaluate the performance of the method. We consider all keywords that appear at least 5 times in the selection set and at least once in the validation and test set, respectively. This results in 59 keywords. The validation set is composed of 697 characters and the test set of 696 characters.

For parameter optimization, we compare two scenarios: validation with real data (fully annotated) and validation with printed characters to avoid human annotations (font validation).

- **Fully annotated.** The validation is done with the real characters from the validation set.
- **Font Validation.** The validation is done with a synthetic validation set that consists of printed characters. 20 random keywords are chosen and printed in 5 Chu Nom fonts, leading to 5 templates per keyword. The synthetic validation set is then completed to a total of 1,000 characters with 900 random non-keyword characters, each printed randomly with one of the fonts.

Only a single hyper-parameter configuration is considered for character detection with YOLO: The default configuration of the medium-sized YOLOv5m model[3] with weights pretrained on the COCO dataset. The deep convolutional neural network is trained until convergence over 25 epochs on 30,000 synthetic pages with printed Chu Nom characters (see Section 3.1) with an initial learning rate of 0.0032.

The detected character images are normalized to a width of S pixels (see Section 3.2) and the node labels of the extracted graphs are normalized to zero mean and unit variance (z-score) to remove small differences in position and stroke length.

The optimization of the hyper-parameters of the graph extraction and graph matching was conducted in two steps.

- **1st step**. We tested 3 different widths $S \in \{50, 100, 150\}$ in combination with three node distances $D \in \{5, 10, 15\}$ pixels to explore different degrees of super-resolution. We tested 3 combinations of node and edge costs $c_V, c_E \in \{0.5, 1.0, 1.5\}$. The largest width S with the smallest node distance D gave the best results.
- **2nd step**. The width S was tested with $S \in \{100, 150, 200\}$ and the node distance with $D \in \{3, 4, 5\}$ pixels, keeping the same combinations of node and edge costs $c_V, c_E \in \{0.5, 1.0, 1.5\}$.

The result of the two steps were merged to a three-dimensional visualization in Figure 4, showing the mAP results on the validation set for different character widths S and node distances D. For each point (S, D), the best mAP value among all tested combinations of node and edge costs is indicated. In this figure, we observe that on the validation set composed of real characters, the smaller the distance D and the higher the resolution S of the image, the better the mAP. For reasons of computational time we did not investigate a size of D lower than 3 or characters with a width higher than 200 pixels. For the validation with printed characters, the figure clearly shows a plateau that peaks with a distance D equal to 5 and a character width at 150. Below or above these two values, the mAP is lower.

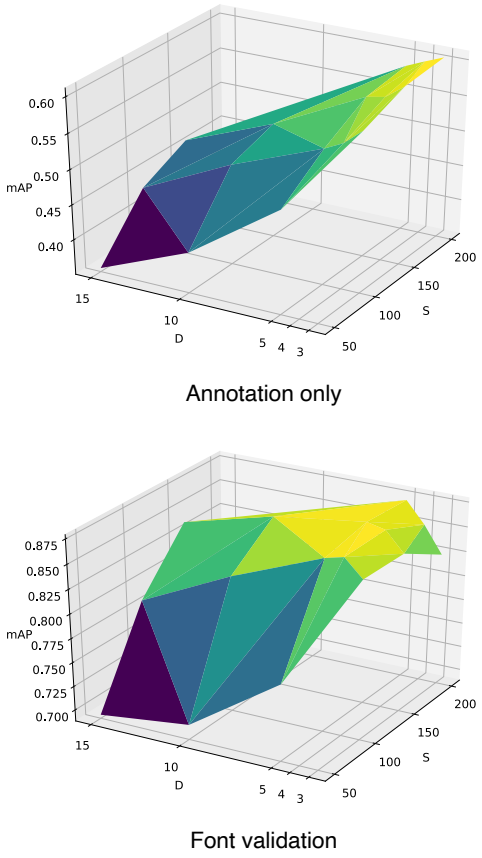

Figure 4: Optimization of character width S and node distance D.

The optimization of node and edge costs (C_e, C_v) for the HED (Figure 5) is very similar between the validation with real characters and the validation with the printed characters. In both cases, the best result was obtained with the highest value of C_e and the smallest of C_v. Additional experiments with printed characters using even

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

higher values of C_e and smaller values of C_v did not change the results significantly.

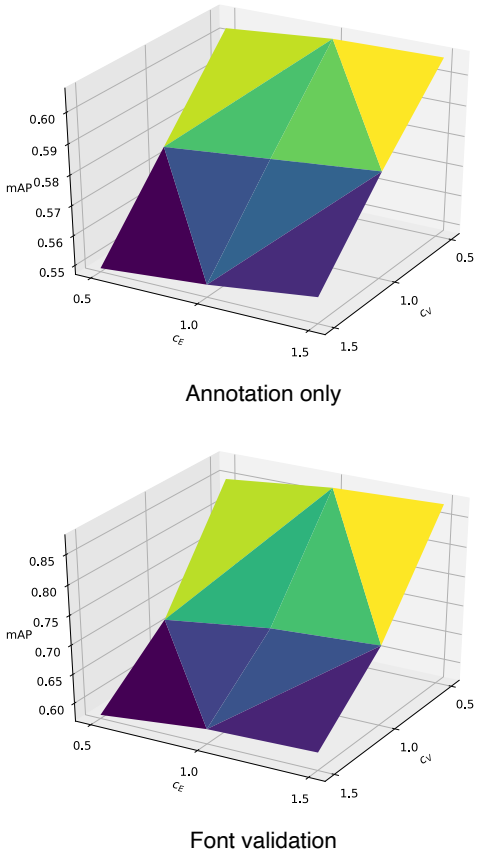

Figure 5: Nodes and edges costs optimisation (c_V, c_E) for HED.

Table 1 shows the best parameters for the two different settings, i.e. using real characters (fully annotated) and using printed characters (font validation). They are quite similar in both cases.

Table 1: Hyper-parameters after optimisation.

Parameters	Fully annotated	Font validation
Width S	200	150
Node dist. D	3	5
Node cost c_V	0.5	0.5
Edge cost c_E	1.5	1.5

Finally, Table 2 shows the mAP on the test set of steles. The performance level is similar between optimization on real and printed data, with only a slight decrease in performance when using printed characters. For the remainder of the experiments, we have fixed the parameters to those of the font validation to make sure no human annotations were used during hyper-parameter optimization.

Regarding computational time, the HED is efficient with a quadratic time complexity $O(n^2)$ with respect to the number of graph nodes n. A few milliseconds were needed to match two keypoint graphs and the duration of a complete keyword spotting experiment with fixed parameters was about one hour. Nevertheless, to investigate all parameter combinations discussed in this section, we had to use a cluster environment with hundreds of computational nodes in order to perform the experiments in parallel.

Table 2: Mean Average Precision (mAP) on the test set after optimization of the parameters.

	Test set
Fully annotated	0.61
Font validation	0.58

4.2 Spotting setup

We perform the final evaluation of the keyword spotting system with respect to a setup that aims at being as close as possible to a real use case, using all characters at our disposal. In this setup, we use the whole set of available characters as a spotting set, minus the 59 keyword characters times their 5 templates. Two scenarios are distinguished:

- **Font validation.** We use the best hyper-parameters of the font validation but extract the character images according to the ground truth bounding boxes.
- **Annotation-free.** We use the best hyper-parameters of the font validation and rely on the automatic character detection to extract the character images. Hence, this scenario is completely annotation-free.

For the font validation scenario, the resulting spotting set is composed of 2,491 characters and for the annotation-free scenario, the spotting set contains 2,312 characters. The difference is due to the fact that not all characters were automatically detected.

Regarding the keyword templates, we evaluate both the query by example setup, which relies on 5 real character templates, and the query by string setup, which considers 5 printed character templates based on 5 different fonts.

Among the characters to be spotted, 3 characters are of particular interest to the historians for the study of the steles: 錢 (tien), 社 (xã), and 為 (vi), respectively. The numbers after the character 錢 indicate amounts of money. By searching for the two characters before 社, a list of village names can be accessed. Finally, the reason for donations is most of the time introduced by the character 為.

4.3 Spotting Results

Table 3 presents the spotting results obtained for the steles and puts them in relation to the spotting results reported in [14] for historical manuscripts containing the tale of Kieu.

In both cases, the spotting performance on ground truth bounding boxes (font validation) is better than the performance on automatically detected characters (annotation-free). The difference is more important in the case of the steles, where we have to take into account a higher number of errors when detecting the characters. Surprisingly, the system is also able to perform query by string keyword spotting using printed character templates, although the mAP

is significantly lower for the more difficult case of stone engravings when compared with manuscripts.

When compared with the state of the art, mAP results of over 90% were reported for historical Latin manuscripts when considering perfect (manual) word segmentation and training deep convolutional neural networks with a large number of human-annotated learning samples [17]. Although our approach results in a lower mAP, it is important to highlight that the achieved results are still very promising, because our method performs a fully automatic character segmentation and does not use any human-annotated learning samples.

Table 3: Mean Average Precision (mAP) on the spotting set of the steles, compared with the results obtained for manuscripts (Kieu dataset [14]).

	Manuscripts	Steles
Font validation	0.78	0.51
Annotation-free	0.77	0.42
Query by string	0.63	0.35

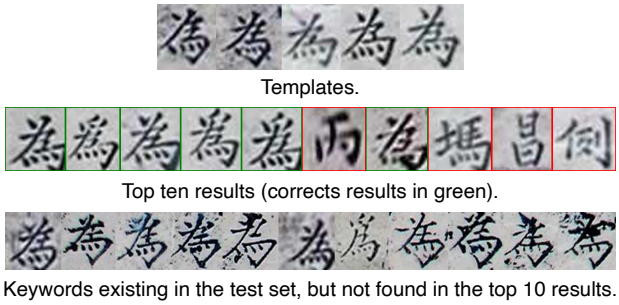

Figure 6: Character spotting (vi).

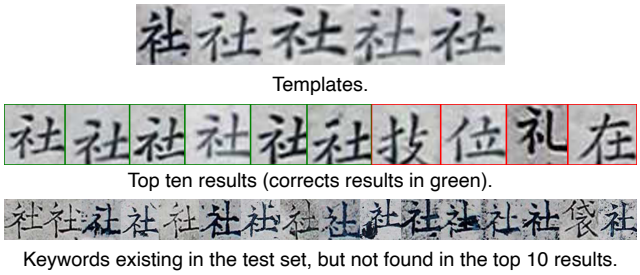

Figure 7: Character spotting (xã).

Figures 8, 7 and 6 illustrate three examples of keyword searches, for the characters 為 (vi), 社 (xã), and 錢 (tien). On the first line, the five template images of the keyword are shown, illustrating the different styles covered by the templates. On the second line, the 10 best results are displayed according to their spotting score (Equation 2), with correct spotting results highlighted in green and incorrect results marked in red. The last line contains the keywords

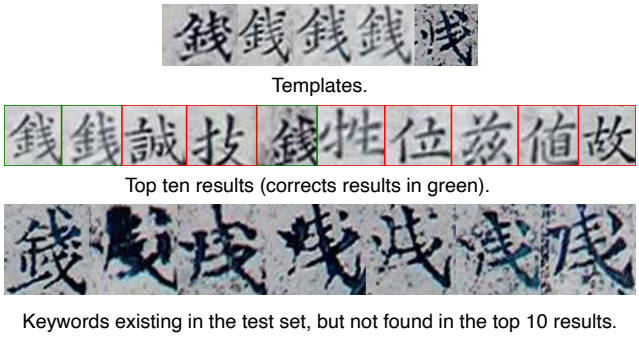

Figure 8: Character spotting (tien).

present in the spotting set but not present in the 10 best results. We notice that the characters, which are correctly recognized, are in the top ranks. They have a style close to the template characters and contain little or no noise. On the other hand, the missed characters contain noise and/or have a different style than the templates.

5 CONCLUSIONS

We have investigated a complete pipeline for keyword spotting in historical Vietnamese steles that does not require any image preprocessing or human-annotated learning samples. By combining self-calibrated, convolutional character detection networks with a graph-based representation of the Chu Nom characters, we have demonstrated that keywords can be retrieved with promising precision for well-readable characters, whose engraving style is represented in the template images.

In order to deal with noise in the character images and a larger number of engraving styles, we envisage several lines of future research. First, geometric deep learning would be interesting to investigate for learning the styles using graph-based representations [11]. Data augmentation may be pursued to integrate more robustness to noise. Finally, it may be possible to extend the annotation-free approach to other scripts and languages.

ACKNOWLEDGMENTS

This work has been supported in parts by the Vietnamica project (ERC Advanced Grant 833933).

REFERENCES

[1] H. Bunke and G. Allermann. 1983. Inexact Graph Matching for Structural Pattern Recognition. *Pattern Recognition Letters* 1, 4 (1983), 245–253.
[2] Andreas Fischer, Marcus Liwicki, and Rolf Ingold (Eds.). 2020. *Handwritten Historical Document Analysis, Recognition, and Retrieval — State of the Art and Future Trends.* World Scientific.
[3] A. Fischer, K. Riesen, and H. Bunke. 2010. Graph Similarity Features for HMM-Based Handwriting Recognition in Historical Documents. In *Proc. Int. Conf. on Frontiers in Handwriting Recognition.* 253–258.
[4] A. Fischer, C. Y. Suen, V. Frinken, K. Riesen, and H. Bunke. 2015. Approximation of graph edit distance based on Hausdorff matching. *Pat. Rec.* 48, 2 (2015), 331–343.
[5] Thai V. Hoang, Salvatore Tabbone, and Ngoc-Yen Pham. 2009. Extraction of Nom Text Regions from Stele Images Using Area Voronoi Diagram. In *10th International Conference on Document Analysis and Recognition.* 921–925.
[6] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 8759–8768.
[7] Paul Maergner, Vinaychandran Pondenkandath, Michele Alberti, Marcus Liwicki, Kaspar Riesen, Rolf Ingold, and Andreas Fischer. 2019. Combining graph edit

distance and triplet networks for offline signature verification. *Pattern Recognition Letters* 125 (2019), 527–533.

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

[9] Philippe Papin. 2003. 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 (2003), 465–472.

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

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

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

[13] Anna Scius-Bertrand, Michael Jungo, Beat Wolf, Andreas Fischer, and Marc Bui. 2021. Annotation-Free Character Detection in Historical Vietnamese Stele Images. In *Proc. 16th Int. Conf. on Document Analysis and Recognition (ICDAR)*. 432–447.

[14] Anna Scius-Bertrand, Linda Studer, Andreas Fischer, and Marc Bui. 2022. 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*.

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

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

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

[18] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. 2020. 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*. 390–391.