Keywords spotting in Russian handwritten documents based on strokes segmentation

Dmitrii D. Feoktistov[1], Leonid M. Mestetskiy [1, 2]

[1] Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, 119991 Moscow - feoktistovdd@my.msu.ru, mestlm@mail.ru

[2] National Research University Higher School of Economics (HSE University), 109028 Moscow

Keywords: Keyword Spotting, Document Image Processing, Strokes Segmentation, Fourier Descriptor

Abstract

The keywords spotting task in handwritten documents is as follows: a user enters text that needs to be searched for in a corpus of handwritten documents. This task can significantly simplify work with archived data. We propose a two-stage algorithm to solve this problem. The first stage involves classifying the strokes, which are the main elements of handwriting. To do this, a measure of similarity based on a Fourier descriptor for elements of the stroke representation is proposed. The second level of the algorithm involves matching the query with the text. An algorithm based on optimal string alignment distance is used for this purpose. To demonstrate the results and adjust the parameters of algorithm we use images of works completed during ”Total Dictation” exam.

1. Introduction

Algorithmic technologies of search engines play a significant role in the modern world. They have become a universal tool used in various tasks related to large volumes of text data. These search engines provide efficient searching and ranking of words and text fragments within any text file repository. However, not all text archives are in a text file format. There are also literary and historical archives containing handwritten texts, which are stored in the form of digital images obtained through scanning or photographing paper documents. These archives consist of a vast amount of data, with volumes reaching hundreds or thousands of pages, representing the cultural heritage of states, peoples, and prominent writers and scientists. Despite this vast collection, a significant portion of these archives remains unexplored due to the complexity of working with handwritten images. It is carried out by small groups of researchers, literary experts and historians who analyze and study these materials, making them accessible to the public. One of the ways to speed up the work of these groups is through handwritten text recognition. This is an active area of research, however the quality of text recognition is not yet at a level where it is possible for direct reading to take place (AlKendi et al., 2024). Even when there is a transcription of documents in the form of text files, researchers still work with digital images because the handwriting, format, structure, and texture of manuscripts contain information that is not present in text transcriptions.

The experience of creating and developing search engines, specifically Google and Yandex, demonstrates the revolutionary role that automation of search queries based on keywords and phrases plays in modern society. It is reasonable to assume that the development of similar technology for text archives presented in the form of manuscripts would have an equally significant impact on the accessibility, study, and preservation of cultural heritage.

When working with handwritten archives, researchers often need to search for specific words, phrases, letters, lines, or text formats. They also need to find information about the location of text on pages. With the help of spotting tools, researchers can easily index data and create their own indexes for people, places, events, and more.

Algorithms for keywords spotting in handwritten documents play a crucial role in the development of search tools. The relevance of automating the processing of handwritten archives using search queries depends on these algorithms.

From a technical perspective, there are several approaches to searching for handwritten text. In the first approach, the user provides an example of a desired word, which allows the system to bypass the challenge of handwriting variability. A more general method assumes that the user inputs a string of text. Also there are several ways to tackle keywords spotting task. The first approach involves using deep neural networks directly. The seccond one involves converting images of words into embeddings, which can then be searched using machine learning techniques. This can be done using discrete representations of words (Sfikas et al., 2017) (Kundu et al., 2021), or continuous features derived from skeletal graphs (Ameri et al., 2017) (Stauffer et al., 2016). The latter approach is still relevant today, as there is an experimentally confirmed hypothesis that increasing the number of parameters in neural networks does not necessarily lead to improved search results (Rusakov et al., 2018). Therefore, it can be concluded that classic methods should also be explored to solve the keywords spotting in handwritten documents task, as neural networks alone may not be sufficient.

The proposed approach builds upon classic keywords spotting methods for handwritten text and based on a novel technique called stroke segmentation (Mestetskiy, 2023). This technique reflects the natural process of writing and offers a new perspective on the problem.

2. Problem statement

The search task, in its classical form, is to establish a correspondence between a collection of data called a file and an individual piece of data called a sample or query. This involves determining the location of the sample within the file.

In relation to the task of keywords spotting in a handwritten text, the input file is an archive of digital images of handwritten

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-2-W5-2024-49-2024 | © Author(s) 2024. CC BY 4.0 License.

pages. The query can be a word or a phrase, and the search output includes two main components. Firstly, the output provides the list of all pages where the desired keyword or phrase appears, along with the exact location of each occurrence. This is achieved by highlighting the relevant context in the image of each page. Secondly, the search result ranks the found occurrences based on their relevance. This ranking is determined by a quantitative assessment of how accurately the search engine has identified the desired occurrence.

The search process is carried out by sequentially examining images from corpus and measuring the similarity between the query and these fragments. Developing an appropriate metric for comparing these objects is a crucial part of search algorithms. In order to create such a metric, it is necessary to map these objects into a common metric space.

In this setting, the natural measure of quality is $accuracy@k$ – the proportion of queries for which the answer is among the k nearest neighbors.

3. Method description

3.1 Method scheme

In this work we describe an approach to building a common space for queries and text files, as well as the distance function between them. The proposed solution is based on representing handwritten text as a sequence of strokes – basic calligraphic elements created by the lines drawn by a pen on paper. Thus, handwritten text images are transformed into tuples of strokes forming a metric space. The metric is based on the optimal string alignment distance between tuples, which are constructed via measurement of the distances between individual strokes. The distance between strokes is based on representation of strokes as broken lines and calculation of Fourier descriptors for such lines. Construction of a stroke representation from an image of handwritten text is based on reconstruction of pen trajectory. Method of its solution described in (Mestetskiy, 2023).

Therefore, the proposed approach involves five stages:

1. Converting a request into an image.
2. Building a stroke segmentation for an handwritten text image.
3. Strokes normalization.
4. Strokes classification based on the Fourier descriptor.
5. Calculation of the modified optimal string alignment distance for sequences of stroke classes.

In the following sections, we will discuss each stage in more detail.

3.2 Converting a request into an image

The formation of a search query in handwritten text depends on the origin of the text file. If the file contains texts by different authors, their handwriting styles may vary greatly. In such cases, the query is formatted in a universal way. For example, it can be written in the Propisi handwritten font (Figure 1). If the search is performed on a text written by a single author, the query can be created using the author’s handwriting or by simply selecting a keyword from the file. In all cases, the intermediate result is a bitmap image of the query in the form of handwritten text.

Figure 1. Stroke segmentation of a query.

Further processing of this image involves converting it into a stroke representation using the same algorithm as described in the next section for handwritten text images (Figure 1).

3.3 Stroke segmentation

Segmentation is performed based on a continuous medial representation of a digital binary image. This medial representation (Mestetskiy, 2023) consists of a skeleton formed by the centerlines and a radial function that describes the width of the object with respect to the points on the skeleton. To create the skeleton, the binary image of the text is approximated using polygonal shapes – polygons with polygon holes. The skeleton is composed of a set of centers for all circles inscribed within the polygonal shapes, forming a loaded geometric graph. Each vertex in the graph represents a point on the plane, while each edge represents a line connecting two vertices. The edges of the skeleton can be straight lines or quadratic parabolas, although when applied to handwritten text processing, they can often be accurately approximated as polylines.

The process of creating a continuous medial representation of a binary bitmap image is described in (Mestetskiy, 2023). It includes the following steps (Figure 2):

- Approximation of all connected black components in a binary image (Figure 2a) by polygonal shapes – polygons with polygonal holes (Figure 2b);
- Construction and pruning of the medial representation of polygonal shapes in the form of skeletons consisting of a set of median axes (Figure2c), and radial functions defined on these axes and describing the width of objects respect to the axes. The skeleton is then cleaned of ”noise” branches through a process called pruning.

The stroke representation of handwritten text is created as a sequence of individual strokes, which are subgraphs of a skeleton graph. There are two types of strokes: ring strokes, which represent cycles in the skeleton graph, and branch strokes, which represent chains in the graph.

Thus, strokes are defined by sequences of vertices, which are connected by lines. The direction of a stroke is determined by

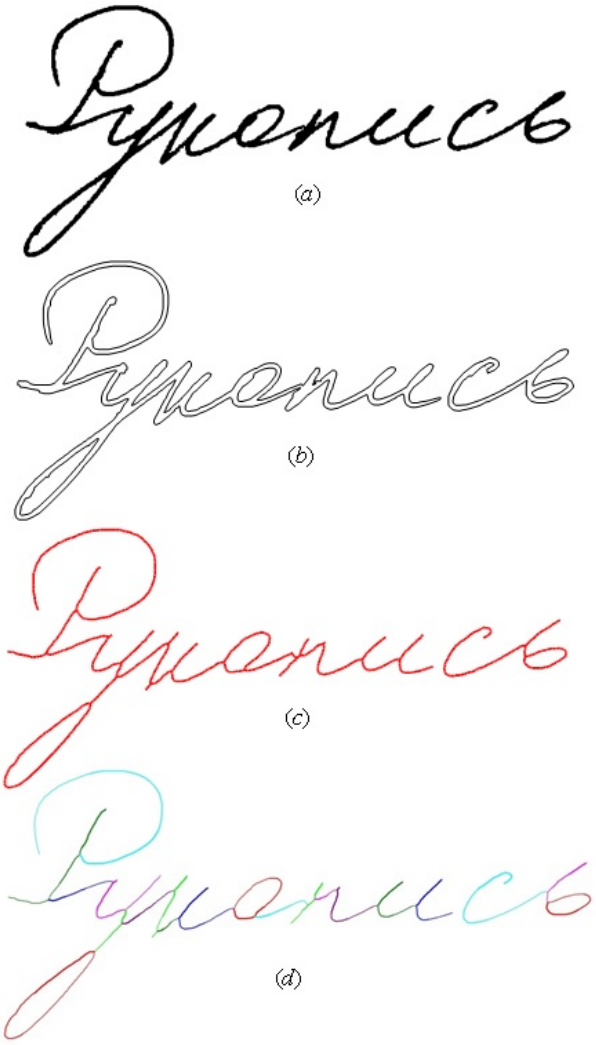

Figure 2. Creation of a continuous medial representation for an image of handwritten text.

the starting point – this is one of the two endpoints of the chain. For rings, in addition to the starting point, the direction of traversal is explicitly set: clockwise or counterclockwise.

The stroke representation is formed by sequentially highlighting cycles and chains that make up ring and branch strokes. This can be thought of as cutting a graph with vertices into smaller subgraphs. The original graph has vertices with degrees of 1, 2, and 3, and the cutting occurs at vertices with degree 3.

Based on the calligraphic origins of the skeletal graph, there are two criteria that are used when making cuts. First, all cyclic chains in the skeletal graph must be preserved. Secondly, it is important to correctly determine which edges incident to a vertex with degree 3 were formed by a single pen movement, and which were formed by an attached stroke.

The definition of ring strokes is based on the property that they consist of edges that separate different components of a plane graph, created by a skeleton. After the ring strokes have been selected, the remaining parts of the skeleton do not have any cycles. To create the branch strokes, the third-degree vertices need to be cut.

To cut the graph at each vertex with degree 3, we need to determine which of the three edges should be removed from this vertex. This problem can be solved by choosing a chain that passes through the vertex and has the least curvature. The curvature estimate can be obtained by approximating the chain with a smooth Bezier curve of second degree.

The choice of the curve with the least curvature is based on the observation that a single cursive stroke is written with a smooth movement of the pen, without sudden changes in direction. If there is a sudden change in direction, it should be interpreted as the end of one stroke and the start of another.

The strokes are arranged based on their relative position. For each stroke, a frame is calculated, which is the smallest rectangle with sides parallel to the coordinate axes that encloses the stroke. The position of the stroke is defined by the center point – the center of the frame. The strokes that lie on the baseline are ordered by their center points, from left to right. Protruding and hanging strokes are placed directly after the baseline stroke with which they share vertices.

3.4 Strokes normalization

After receiving the polylines that describe the stroke, we need to preprocess them in a certain way for further construction of the Fourier descriptor. Let’s say we have a polyline $L = \{(x_j, y_j)\}_{j=1}^{l}$. We propose applying the following transformations to this polyline.

1. Parallel transfer of the polyline describing the stroke, such that the center of mass is located at the origin of the coordinates.

$$(x_j, y_j) := (x_j - \frac{1}{l}\sum_{k=1}^{l} x_k, y_j - \frac{1}{l}\sum_{k=1}^{l} y_k)$$

2. If the stroke is not a circle, the traversal order will be set so that the start point is on the left side of the end point.

3. If the stroke is circular, then the traversal starts from the leftmost point of the stroke in a clockwise direction. The following observation is used to determine the direction: if polyline L is traversed in a clockwise, then it is true that:

$$\sum_{j=0}^{l-1}(x_{j+1} - x_j)(y_{j+1} - y_j) + (x_0 - x_l)(y_0 - y_l) > 0$$

4. To standardize the sizes, all strokes are normalized to the average stroke size in the image. The stroke size refers to the length of the polyline:

$$Size(L) = \sum_{j=0}^{l-1}\sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}$$

5. Then we increase the number of points uniformly along the length of the stroke, until N_{points} dots have been reached.

3.5 Strokes classification

The next step in the algorithm is to classify the strokes. We propose to use strokes Fourier descriptors as features for classification. (Zahn and Roskies, 1972).

Let the stroke be represented by a polyline $U = \{(x_j, y_j)\}_{j=1}^{l}$. It can be mapped to a sequence of complex numbers $U^c = \{u_j^c = x_j + i \cdot y_j)\}_{j=1}^{l}$, for which we perform a discrete Fourier transform:

$$f_l = \sum_{k=0}^{N-1} u_k^c \cdot exp(-i \cdot \frac{2\pi}{N} \cdot l \cdot k) =$$

$$= u_0^c + \sum_{k=1}^{N-1} u_k^c \cdot exp(-i \cdot \tfrac{2\pi}{N} \cdot l \cdot k), \quad l = 0...N-1 \quad (1)$$

Lets associate with the stroke a vector composed of the first N_{coef} Fourier coefficients.

Next, we propose using a metric classification algorithm based on prototypes. In order to do this, we need to answer two questions: which distance should we use and where can we find the prototypes?

It is suggested to use the Euclidean distance between Fourier descriptors x_1, x_2 as a measure of distance:

$$\rho(x_1, x_2) = \|x_1 - x_2\|_2 \quad (2)$$

Or, we can map complex vectors to real vectors $r(x) = (real(x), imag(x))$. Then, we can use the cosine distance:

$$\rho_1(x_1, x_2) = 1 - \frac{\langle r(x_1), r(x_2) \rangle}{\|r(x_1)\|_2 \cdot \|r(x_2)\|_2} \quad (3)$$

To highlight the prototype strokes, we propose to use one of two methods:

- Manual selection of prototypes;
- Selection of reference strokes using the K-means clustering algorithm in the Fourier descriptor space for data representing an image of large text written in the Propisi font. After that, Fourier descriptors of the cluster centroids can be used as a description of the reference strokes. In this case, the number of patterns is denoted by $N_{prototypes}$.

Both approaches produce similar results, which will be demonstrated in the experiments section.

3.6 Modified optimal string alignment distance

Let's match each stroke with the number of the nearest standard. If the stroke is the last one in a word, then we add a special character after its number. We will denote this symbol as -1 for convenience. So the handwritten text can then be described by a sequence of numbers. The optimal string alignment distance (Navarro, 2001) can be used used as a distance measure for this type of data. However, this algorithm doesn't take into account the fact that errors can vary in importance, or that end-of-word character cannot be changed. To address these issues, we propose modified optimal string alignment distance so that it takes these features into account. The resulting recurrent formula for calculating the distance is:

$$d_{a,b}(i,j) = min \begin{cases} 0, & i=j=0 \\ d_{a,b}(i-1,j) + deletion, & i>0,\ j=0 \\ d_{a,b}(i,j-1) + deletion, & j>0,\ i=0 \\ d_{a,b}(i-1,j-1) + rep(a_i, b_j) & i,j>0 \\ d_{a,b}(i-2,j-2) + swap \cdot [a_i \neq b_j], & i,j>1,\ a_i = b_{j-1},\ a_{i-1} = b_j \end{cases} \quad (4)$$

Where:

$$rep(a_i, b_i) = \begin{cases} 0, & a_i = b_j \\ +\infty, & a_i \neq b_j, a_i = -1 \text{ or } b_i = -1 \\ replacement, & \text{else} \end{cases} \quad (5)$$

$deletion$, $replacement$, $swap$ – hyperparameters of the proposed algorithm.

4. Experiments

4.1 Distance evaluation

To begin with, we demonstrate that the distances based on the Fourier descriptor satisfy the compactness hypothesis. By doing so, we will understand that the ratio of the average intra-class distance to the average inter-class distance is less than one, and also that, with its help, the KNN algorithm can solve the problem of stroke classification with high accuracy during cross-validation. As data, we use several sentences containing all the letters of the alphabet, where each of the 1251 strokes has been manually assigned to one of the 12 classes. The prototype strokes of these classes are shown in Figure 3.

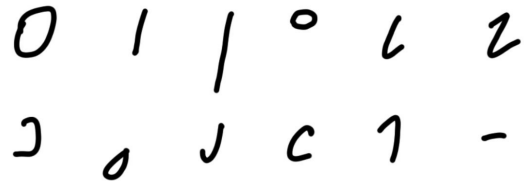

Figure 3. Prototype strokes. Each of them corresponds to one basic calligraphic element.

As can be seen in Figure 4, the compactness hypothesis holds for all quantities of Fourier coefficients. However, as the number of classes increases, the separability between them decreases, which can be attributed to the appearance of coefficients that correspond to noise, as well as to the curse of dimensionality.

For the classification problem, we will use the F_1 macro-averaged measure as a quality metric, since there is an imbalance in the data: the largest class contains 250 strokes, and the smallest contains 28. As can be seen in Figure 5 the proposed distance allows us to distinguish strokes with high quality and stability, regardless of the number of Fourier coefficients used. Additionally, the Euclidean distance performs better at solving this problem.

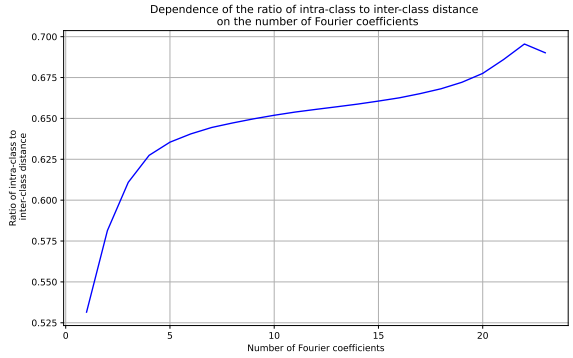

Figure 4. Fulfillment of the compactness hypothesis: The values of the Fourier coefficients are plotted along the x-axis, and the ratio of the average intra-class variance to the average inter-class variance is plotted along the y-axis. As shown in the graph, the ratio does not exceed 1, indicating that the compactness hypothesis is fulfilled.

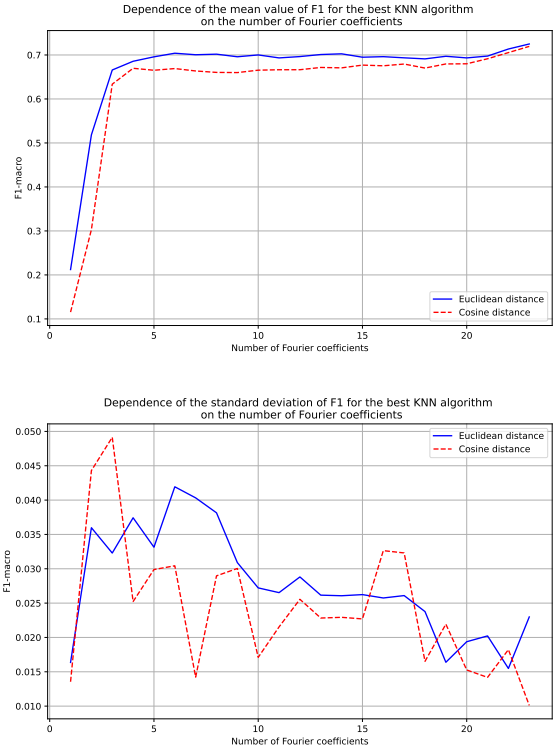

Figure 5. The results of an experiment investigating the quality of stroke classification based on the number of Fourier coefficients. The values of the number of Fourier coefficients used are plotted along the x-axis, and the corresponding F_1 scores are plotted along the y-axis.

4.2 Templates extraction algorithms comparision

In the section 3.5, we described two algorithms for selecting prototype strokes. To compare the proposed algorithms, we conducted an experiment using a text containing $3,695$ strokes printed using the Propisi font. Each stroke was assigned to one of the classes using the nearest neighbor method, with classes obtained either from markup or a clustering algorithm. After that, the adjusted mutual information coefficient (IMU) (Vinh et al., 2009) was calculated for each class. In the experiment, the value of IMU turned out to be 0.57. This indicates that the obtained classes are similar and the algorithm for selecting classes through clustering is meaningful. These conclusions can also be confirmed by visual analysis of the results (Figure 6).

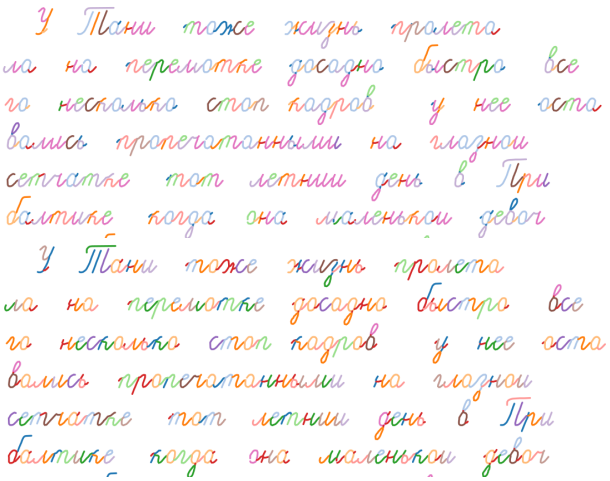

Figure 6. Stroke markup obtained using classification (top) and clustering (bottom).

4.3 Spotting algorithm evaluation

Now we will evaluate proposed algorithm in searching in a handwritten context task. We will use images of works completed during ”Total Dictation” exam. This data includes pictures of works of 4 participants, each with 28-31 lines. The set for searching will consist of lines from these works, and the queries will be printed texts that contain words from these lines. An example of such a search set is shown Figure 7.

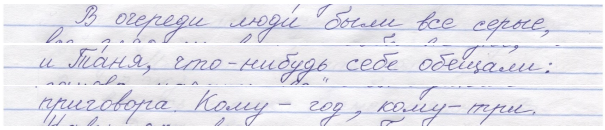

Figure 7. Example of lines included in the search space.

As a result of the experiments, we found the following hyperparameters for the algorithm: $N_{points} = 89$, $N_{coef} = 16$, $N_{prototypes} = 14$, $deletion = 0.62$, $replacement = 0.42$, $swap = 0.0$. The obtained values of the $accuracy@k$ metric are presented in the Table 1.

k	$accuracy@k$
1	0.74
2	0.79
3	0.82
5	0.86

Table 1. The dependence of the accuracy@k metric on k for the resulting algorithm.

5. Conclusion

The paper presents a novel approach to keywords spotting in handwritten documents task based on stroke segmentation. The

proposed algorithm shows high accuracy in discussed problem and demonstrates the potential of using stroke analysis for handwritten text documents processing.

6. Acknowledgements

This work was supported by the Russian Science Foundation, project no. 22-68-00066

References

AlKendi, W., Gechter, F., Heyberger, L., Guyeux, C., 2024. Advancements and challenges in handwritten text recognition: A comprehensive survey. *Journal of Imaging*, 10(1), 18.

Ameri, M., Stauffer, M., Riesen, K., Bui, T., Fischer, A., 2017. Keyword spotting in historical documents based on handwriting graphs and hausdorff edit distance. *International graphonomics society conference*, 105–108.

Kundu, S., Malakar, S., Geem, Z. W., Moon, Y. Y., Singh, P. K., Sarkar, R., 2021. Hough transform-based angular features for learning-free handwritten keyword spotting. *Sensors*, 21(14), 4648.

Mestetskiy, L., 2023. Stroke segmentation of handwritten text. *Reports of the All-Russian conference ”Mathematical methods of pattern recognition” (MMPR-2023), FRC CSC RAS, Russia*.

Navarro, G., 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1), 31–88.

Rusakov, E., Sudholt, S., Wolf, F., Fink, G. A., 2018. Expolring architectures for cnn-based word spotting. *arXiv preprint arXiv:1806.10866*.

Sfikas, G., Retsinas, G., Gatos, B., 2017. A phoc decoder for lexicon-free handwritten word recognition. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01, 513–518.

Stauffer, M., Fischer, A., Riesen, K., 2016. Graph-based keyword spotting in historical handwritten documents. *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, S+ SSPR 2016, Mérida, Mexico, November 29-December 2, 2016, Proceedings*, Springer, 564–573.

Vinh, N. X., Epps, J., Bailey, J., 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? *Proceedings of the 26th annual international conference on machine learning*, 1073–1080.

Zahn, C. T., Roskies, R. Z., 1972. Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3), 269–281.