A Character Style Library for Syriac Manuscripts

Nicholas R. Howe
Smith College
Northampton, MA (USA)
nhowe@smith.edu

Alice Yang
Smith College
Northampton, MA (USA)
ayang@smith.edu

Michael Penn
Mt. Holyoke College
South Hadley, MA (USA)
mpenn@mtholyoke.edu

ABSTRACT

Paleographers study ancient and historical handwriting in order to learn more about documents of significant interest and their creators. Computational tools and methods can aid this task in numerous ways, particularly for languages and scripts that are not widely known today. One project currently underway seeks to gather a collection of securely dated letter samples from Syriac documents dating between 500 and 1100 CE. The set comprises over 60,000 human-selected character samples. This paper gives details on the collection and describes the automatic techniques used to process the initial human input so as to produce high-quality segmented character samples ready for analysis.

CCS Concepts

•**Applied computing** → *Digital libraries and archives;* Arts and humanities;

Keywords

Syriac, historical manuscripts, handwriting style

1. INTRODUCTION

The study of handwriting can offer significant insight into historical manuscripts and the people who created them. Handwritten forms vary with time and location, and thus offer useful clues to a document's provenance. The study of ancient and historical manuscripts to extract such information is known as paleography.

Human scholars have long relied on personal expertise to attribute and date manuscripts. The use of computers for this task is more recent but has yielded tantalizingly promising results [?, ?, ?]. Automated methods offer a particular advantage for languages that have received relatively little human attention to date, since the answers to more questions still await discovery.

Syriac is a language that appears ripe for investigation. While modern variants of Syriac are still spoken in a few isolated communities today, the language was widely used by scribes of religious texts in parts of what are now Turkey, Syria, Iran, and Iraq from 200 CE onwards. Tens of thousands of these documents have been preserved in libraries and collections around the world. Most of them lack secure dates, and thus cannot reveal as much about their role in history as one might desire. Tools that can better help scholars to place undated manuscripts in context would therefore be of great value.

As a first step towards this goal, we have gathered copies of a large fraction of the securely dated Syriac manuscripts known to exist in the world today. In most cases, we know when these artifacts were created because the scribe included a colophon with a date or other temporal clue. Such documents can serve as signposts or landmarks from which the dates of other manuscripts may be estimated by comparison. Nonetheless, the features that make such comparisons feasible are not necessarily immediately apparent to the layperson or accessible to a computer algorithm. This paper leaves for future work the creation of an automated dating system. However, it describes the processing steps taken to make the differences between documents more readily apparent and thus more easily comparable. The number of character samples involved necessitates an approach that is at least semi-automated, and this is the procedure described herein.

1.1 Related Work

Several research groups have looked at automatic techniques for making paleographic judgments for dating or other purposes. Bulacu and Schomaker have proposed a handful of script-independent document statistics which can discriminate between different handwriting styles in testing on medieval documents [?]. More recently, He et al. have worked specifically with medieval Latin and developed techniques based upon the temporal evolution of individual character forms [?], somewhat as intended here.

A group working with ancient Hebrew texts has looked at writer identification based on handwriting style [?], and also presents several paleographic techniques on the restoration and isolation of characters in badly damaged documents [?, ?]. The latter method employs active contours, which perform a function similar to the part-structured models used here. The difference is that active contours require iterative optimization, while part-structured models permit analytical computation of the optimal fit.

A number of other efforts have built libraries of historic manuscripts in various languages [?]. Although not necessarily specifically focused on paleography, such efforts often aim

HIP 2015 August 22, 2015, Nancy, France
© 2015 ACM. ISBN 978-1-4503-3602-4/15/08…$15.00
DOI: http://dx.doi.org/10.1145/2809544.2809556

for broad scholarly applicability and may include character-level annotations. Such collections hold the potential for paleographic initiatives. However, to the authors' knowledge no other such project is currently underway for Syriac.

2. DATA

Any paleographic effort requires access to a large collection of manuscripts for calibration. Of the 188 securely dated manuscripts written before 1100 CE known to exist today, the collection assembled for this project includes samples from 122, and more are being added as they become available. Because the manuscripts are spread across many different libraries, negotiating access rights takes time and effort. Redistribution rights are often harder, but efforts to secure them have been undertaken so as to make at least a portion of the data set available to the public through the author's web site.

From the start, a mix of human input and automated techniques has been used to process the data. Humans provide valuable oversight and quality control, and can perform steps that present computational challenges. On the other hand, with over 60,000 individual character samples, the size of the data set precludes detailed manipulation of each one by hand.

In particular, the method for identifying sample characters provides a good example of symbiosis between human and computer. While automated character detection methods exist, they show unacceptably high error rates [?, ?]. More subtly, relying on an automated detector may skew the distribution of identified characters toward those most amenable to the detection method used. Because this could bias any subsequent conclusions that might be reached, human control over the detection step appears crucial. However, full annotation of the documents is impossible within a reasonable budget.

With these considerations in mind, trained human agents are used to identify selected character samples in each text using a minimum of effort. A custom annotation engine written in Java displays pages and offers an interface for recording and editing character locations. A simple click-and-drag operation serves to identify a bounding box around each annotated character. Most require no further action, although erroneous markings may be edited or deleted as necessary. Future work will consider moving to a publicly available annotation engine such as Aletheia [?].

The collection so far includes 190 documents, with and without secure dates, including on average around eight pages from each source. The Syriac alphabet has 22 letters, and human agents have identified between ten and twenty instances per character for each document. Eleven of the letters occur in a single variant. Ten have two forms that may be specifically distinguished during annotation (e.g., final form vs. mid-word). One (*taw*) has three forms.

Annotators are instructed to choose bounding boxes that enclose the entire letter tightly. In most cases, this requires choosing boundaries that also include portions of other letters. Syriac writing is cursive in style, so it is common for strokes belonging to other letters to touch or connect to the letter of interest. It would be useful if the human agent could specify the exact boundary of the letter of interest, perhaps by tracing it with a mouse or through some other means. Unfortunately, this would greatly increase the time of interaction required for each sample, making the work prohibitively expensive. Thus automated methods must perform the fine segmentation of character samples within the human-specified bounding boxes. The next section considers different means for doing so.

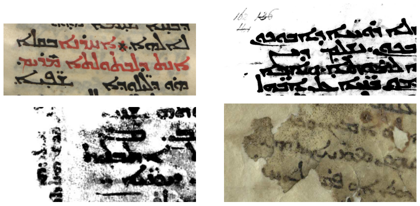


Figure 1: Samples of some documents from the collection, hinting at their diversity. Images ©The British Library Board: Add. 12,148, f. 129a; Add. 14,490, f. 162a; Add. 17,170, f. 5a; Add. 17,213, f. 5a.

3. METHODS

Prior work in computational paleography sometimes glosses over segmentation issues. In languages whose characters are typically isolated from one another by whitespace, this is a viable approach. But Syriac is a cursive script with connected characters, and thus demands more concerted segmentation. Because any subsequent paleography will require reliable input, adequate performance must be constantly ensured. The evaluation may be qualitative in many cases since ground truth is not available, but different approaches can nevertheless be compared. In some cases, postprocessing can be used to reveal problem areas that can be addressed via human intervention or through tweaks to the algorithms used.

3.1 Binarization

Although some paleography techniques may work with grayscale or color images, others require binary images as input, where each pixel is classified as either ink or whitespace. Document image binarization has received ample research attention which can only be alluded to here [?]. However, this project must address several aspects which many papers on binarization do not regularly consider. Primary among these is the heterogeneity of the document sources: they are scanned under many different resolutions and illumination conditions, captured in varying file formats including some with artifact-inducing lossy compression standards, and represent a wide range of preservation quality. Figure ?? shows a few examples to illustrate the variety.

With widely varying documents, appropriate parameter selection becomes crucial. No single setting will work well for all images. Rather, approaches that adapt to their input by automatically selecting appropriate parameter values will have an advantage. This project uses Howe's binarization method, which tunes parameter values using a stability criterion [?]. It is not intended to require any manual adjustment, although as explained below this data set requires a presmoothing step that introduces a new parameter not considered in the original algorithm. Fortunately, sensible values for the smoothing radius can also be set automatically. The resulting procedure works well on a high percentage of input documents, with a few well-understood failure cases. These are rare enough to be detected and addressed

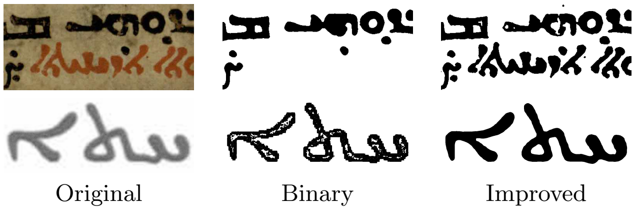


Figure 2: Problems with binarization. First row shows red text (©The British Library Board: Add. 17,256, f. 10a), second row shows high-resolution faded text (simulated example).

by hand.

The need for modifications to the algorithm are apparent from trial runs of Howe's binarization, with several observed problems illustrated by the examples in Figure **??**. The first concerns colored document images: some documents are printed in a mixture of red and black ink, at times together on the same line. Standard conversion to grayscale assigns the red ink a much lighter intensity than black ink. This causes trouble because most binarization algorithms treat faded ink as likely to be a false marking, possibly the result of bleed-through or staining. The binarized images therefore systematically omit most words in red ink while retaining the words in black surrounding them. Although manual adjustment of the binarization parameters can address this issue, the current workaround is to select samples only from areas written in black ink. Since red is used relatively sparingly, this should pose no significant obstacle.

A second problem is more fundamental, and reveals a shortcoming of the competition datasets used to evaluate many of the leading binarization algorithms. Most of the evaluation documents from organized contests contain characters composed of thin strokes at low to medium resolution [**?**], meaning that large areas of pure ink rarely appear. Some of the handwriting examples contain ink strokes with thickness of just a single pixel. On the other hand, Syriac manuscripts tend to be printed with bold, thick lines that are many pixels wide, and algorithms that excel on the test sets described above do not necessarily perform as well here. In particular, a subset of the documents has been scanned at a higher resolution than others, increasing the size of these areas in pixel measurements. If there is any variation to the ink intensity, or the images have been compressed in a way that leaves artifacts, then binarization methods designed for thin strokes will tend to detect smaller-scale structure where none exists. Figure **??** shows the problematic result.

Since high resolution provides the trigger for these bad binarizations, decreasing the resolution should provide a way to address them. One approach would be to downsample the images directly, but this might impact subsequent operations. A better strategy is to keep the original pixel resolution but smooth the image at an appropriate scale so as to suppress the small-scale noise that misleads the binarization. The only question remaining is to find the appropriate scale for smoothing any given image.

We adopt an *ad hoc* approach that seems to work sufficiently well, although perhaps could be refined in future work. It begins with Otsu's binarization method [**?**], which is too simple for a final binarization but gets enough right to provide a starting point. From the Otsu binarization (perhaps with large obviously non-text components thrown out), compute the modal stroke width as an indicator. The smoothing radius is set to some fraction of the measured stroke width, one eighth in the present case. We find the stroke radius by computing a histogram of distance to the ink boundary for points on the medial axis of the Otsu foreground and taking the modal value. Smoothing with a Gaussian filter of this radius fixes most of the binarization artifacts visible in the original. There remain a few cases where too much smoothing is performed, causing loops in some letters to be filled in. Such errors may be caught and corrected in subsequent processing stages, where character models can detect them. Alternately, future work may examine alternative means for determining the best smoothing radius.

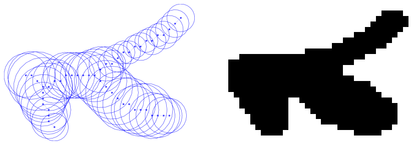


Figure 3: An inkball model (left) and its image rendered in low resolution (right).

3.2 Masking with Inkball Models

To ensure accurate measurements for paleographic applications, each character sample must be masked or segmented to remove the portions of other characters that intrude into the bounding box. The goal is to produce an image of the character alone, without any extraneous markings. Prior work has sometimes relied on simple heuristics, such as removing any disconnected peripheral components [**?**]. Unfortunately, connected characters are common in Syriac, so merely removing the disconnected elements still leaves many extraneous marks intact.

We adopt a model-based approach to character masking. A good model must be flexible enough to adapt to and preserve variations in character form, since it is precisely these difference that will be measured for paleographical purposes. On the other hand, it must not be so flexible as to allow inclusion of foreign strokes within the character mask. Part-structured inkball models [**?**] offer exactly this mix of properties: they are composed of a fixed number of parts, with some flexibility in the linkages that allow adaptation to variations in shape.

A part-structured inkball model can be built from any sample instance of a character by placing maximal disks of ink at regularly spaced points along the medial axis. If the disks overlap sufficiently, they produce a close approximation of the original shape, as shown in Figure **??**. For flexibility, neighboring disks are allowed to shift their position relative to one another according to a Gaussian potential. The resulting template easily matches small deformations in character shape, while penalizing drastic changes. A dynamic programming computation related to the Viterbi algorithm can efficiently compute the lowest-energy configuration of a particular model in relation to a particular observed character sample. The outline of the model's shape then gives a custom mask for the observation. Figure **??** shows some examples.

The model configuration will not necessarily match the

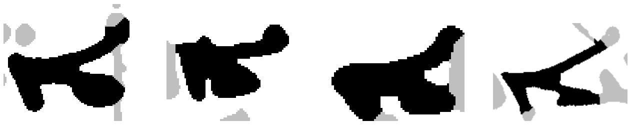
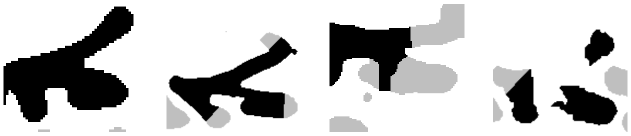


Figure 4: Some masked samples. Area within mask is dark; background ink is shown in a lighter shade.



Inclusion Omission Major Binarization

Figure 5: Examples of various masking and binarization errors.

observed character perfectly, particularly at the edges, so some heuristic postprocessing cleans up the mask. Denote the character sample B, the initial rendered model M_0, and S the medial axis (or skeleton) of B. The final mask M consists of the initial mask plus C, the set of any pixels that are closer to a skeleton point in M_0 than to one outside.

$$(B \cap M_0) \cup C \tag{1}$$

$$C = \{p | p \in B \setminus M_0 \wedge \min_{q \in S \cap M_0} d(p,q) < \min_{q \in S \setminus M_0} d(p,q)\} \tag{2}$$

Here d is the quasi-Euclidean geodesic distance computed within B. Figure **??** illustrates some typical results.

Qualitative evaluation of masking results for the flexible inkball model shows that it does a good job of filtering out stray parts of other characters. The most common error is omission of one or more strokes at the tip. Visual inspection of a representative set of samples shows perfect or near-perfect segmentation in a majority of cases. Section **??** describes the evaluation in more detail. Figure **??** shows an examples of each category of error.

Although the evolution of letter forms tends to be gradual and subtle in many cases, some letters exhibit two or more alternate forms without any transitional examples that bridge the gap. To address these cases we use multiple models, one per canonical form. Each model attempts to fit the sample, and choice goes to the one with the lowest fitting energy as normalized by the number of inkballs it contains. In every observed case this procedure has chosen the correct model.

3.3 Masking with Boundary Models

Inkball models work well in many cases, but as noted above sometimes fail to cover the full length of an extended stroke because their energy function includes no incentive to do so. This limitation can be addressed in several ways. One could modify the energy function so as to include such an incentive, but this would add complication and might still fail to achieve the desired result. One could postprocess the model-based masks using some heuristic to identify and add missing stroke tips. Finally, one could look for a different sort of part-structured model not vulnerable to the same sort of failure. This section introduces a novel character shape model that produces more reliable segmentations than the flexible inkball models used above.

Conceptually, the new shape model also uses a part-structured formulation. Instead of a ball of ink, each part matches to an image edge (or more specifically an oriented gradient in the image intensity). The model as a whole thus represents a set of edges arranged in some characteristic spatial layout, for example to form the boundary of the target letter. Because the model is structured by parts, it retains the flexible matching characteristic of the inkball model, and also the algorithm to analytically compute the optimal configuration.

Part-structured models share a common energy function for matching. Given the model description Q, spatial configuration information C, and observed image I, the energy E splits into two terms.

$$E(Q,C,I) = E_\xi(Q,C) + \lambda E_\omega(C,I) \tag{3}$$

Of these, the internal deformation energy E_ξ measures how much the model parts are deformed from their default configuration. It is formulated identically as for the inkball model and as described in prior work [**?**].

The observation energy E_ω accounts for the match between the model and the image. For the inkball model, this energy measures the sum of the distance from each model node to the nearest pixel on the ink medial axis. For the boundary model, it is defined as the sum of the distance from each model node to the nearest pixel that matches within a specified tolerance in gradient direction and exceeds a threshold in magnitude.

$$E_\omega(C,I) = \sum_{i=1}^{m} \min_{\vec{p} \in G(I,\theta_i)} \|\vec{p} - \vec{v}_i\|^2 \tag{4}$$

Here v_i is the position of model node i under configuration C, and θ_i is its associated gradient direction. $G(I,\theta_i)$ is the set of all pixels that match θ_i within tolerance τ_θ and magnitude threshold τ_a.

$$G(I,\theta) = \{\vec{p} \in I | \angle(\theta, \nabla_\theta I(\vec{p})) < \tau_\theta \wedge \nabla_r I(\vec{p}) > \tau_a\} \tag{5}$$

$$\nabla_\theta I(\vec{p}) = \tan^{-1} \frac{\nabla_y I(\vec{p})}{\nabla_x I(\vec{p})} \tag{6}$$

$$\nabla_r I(\vec{p}) = \sqrt{(\nabla_x I(\vec{p}))^2 + (\nabla_y I(\vec{p}))^2} \tag{7}$$

Computationally, minimizing the model fit energy proceeds exactly as for the inkball model, except that the set of target pixels for the E_ω calculation differs for each node according to its orientation. The image may be either binary (in which case the gradient magnitude is irrelevant) or grayscale. The experiments here use binary images for better comparison with the inkball results.

A boundary model is easily created from a binary sample character. Select m points at equal spacings along the boundary. Record the gradient direction θ_i at each point, and connect the nodes in a ring. For characters with internal holes or multiple components, repeat the process for each boundary as required, then merge the rings into a tree by joining them at the closest neighboring points. Note that each boundary ring will have a break somewhere, due to the computational requirement that node connections must form a tree structure. Future research might look at modifications to the algorithm that would optimize over complete rings. However, a heuristic workaround to the algorithm provides a similar effect: build a model that traverses the entire border twice around, and take the best loop from the

Table 1: Rubric used for evaluation

Omissions	None
	Minor
	Major
Inclusions	None
	Minor
	Major
Binarization problems	None
	Minor
	Major
Annotation correct	Yes
	No

middle of the sequence. Although the ends may not align exactly, the middle of a double loop is usually unaffected.

4. EXPERIMENTS

Although casual inspection shows that most masks have relatively few problems, it is desirable to better quantify this sense, if only to compare the relative merits of the two proposed models. One possible approach would be to build a collection of human-annotated ground truth masks and compare the results to it. We have avoided this approach for two reasons: it is time-consuming, and potentially not very informative. Experience with ground-truthing and binarization available for segmentation show that the summary statistics that might be computed from such a ground truth, such as F-measure, do a poor job of capturing the semantic importance of any error. Depending on their geometry, a certain pixel error rate may represent an entire lobe gone missing, or simply a collection of small variations along the boundary. Paleography often depends on exactly these sorts of subjective distinctions, which are not captured by F-measure.

As an alternative form of evaluation, a human expert has rated the quality of the computed character masks for a total of 680 samples representing five different Syriac characters chosen to include a variety of different forms. One sample per character is drawn from each available annotated document. The expert sees the background ink with the proposed character mask indicated much as in Figures **??** and **??**, and then judges the quality of the mask using the rubric shown in Table **??**.

Tables **??** and **??** presents detailed results from this evaluation, including just the samples with correct annotations. Averaged over all samples and letter types, the two methods show similar rates of serious error: 10.7% for inkball masks and 10.3% for boundary masks. However, the boundary method avoids minor errors more often, scoring perfect marks 64.6% of the time vs. 40.7% for inkball models. The latter were hurt by a tendency to omit the extremities of letters in some cases. Binarization errors are rare, occurring in just 13.5% of cases, and only 6.0% of those are considered serious. Since the annotation project will identify several samples of each letter from every document, with these error rates it should be possible to get at least one high-quality sample character with good probability for every letter on every document.

Table 2: Summary of evaluation results for inkball masking: Percentage of samples showing either no error or minor errors (inclusive).

	Omission		Inclusion		Binarize		Any	
	No	Min	No	Min	No	Min	No	Min
alaph	36	97	97	100	94	98	30	95
taw	66	91	71	99	81	95	29	86
gamal	62	96	71	96	88	95	29	88
semkath	95	99	96	98	82	88	77	86
sadhe	38	99	91	99	92	97	30	95

Table 3: Summary of evaluation results for boundary masking: Percentage of samples showing either no error or at most minor errors (inclusive).

	Omission		Inclusion		Binarize		Any	
	No	Min	No	Min	No	Min	No	Min
alaph	86	98	86	100	94	98	70	97
taw	91	97	86	98	81	95	61	91
gamal	92	98	81	96	88	95	63	89
semkath	98	99	88	95	82	88	69	82
sadhe	74	99	91	99	92	97	62	94

Since each letter sample is annotated with a known type, binarization errors can be detected and corrected by looking for incorrect topologies. For example, the binarization error in Figure **??** splits the target character into three disconnected components. Since the model *alaph* comprises just a single connected component, the presence of three components in this case indicates an error. Either the sample may be simply thrown out, or binarization might be reattempted with parameters better tuned to this particular image patch. Another sort of binarization error can fill in the internal voids for characters containing loops, such as *semkath*. This sort of topological error can also be easily detected and addressed.

5. CONCLUSION

A combination of human effort and automation provides a reliable yet efficient way to get high-quality character samples from a large collection of documents. Such samples can be put to many uses, from training a character detector to paleography. For this particular project we remain interested in questions of dating and attributions. Future work will look at different techniques for representation and comparison of samples.

6. REFERENCES

[1] E. Dalton and N. Howe, "Style-based retrieval for ancient Syriac manuscripts," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, 2011.

[2] S. He, P. Samara, J. Burgers, and L. Schomaker, "Towards style-based dating of historical documents," in *14th International Conference in Handwriting Recognition*, 2014, pp. 265–270.

[3] L. Wolf, L. Potikha, N. Dershowitz, S. R., and Y. Choueka, "Digital paleography: Tools for historical manuscripts," in *18th IEEE International Conference on Image Processing*, 2011.

[4] N. Bulacu and L. Schomaker, "Text-independent writer identification and verification using textural

and allographic features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 701–717, April 2007.

[5] I. Bar-Yosef, I. Beckman, K. Kedem, and I. Dinstein, "Binarization, character extraction, and writer identification of historical Hebrew calligraphy documents," *Int. J. Doc. Anal. Recognit.*, vol. 9, no. 2, pp. 89–99, April 2007.

[6] I. Bar-Yosef, A. Mokeichev, K. Kedem, and I. Dinstein, "Adaptive shape prior for recognition and variational segmentation of degraded historical characters," *Pattern Recognition*, vol. 42, pp. 3348–3354, 2009.

[7] R. Cohen, K. Kedem, I. Dinstein, and J. El-Sana, "Occluded character restoration using active contour with shape priors," in *International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 497–501.

[8] C. Papadopoulos, S. Pletschacher, C. Clausner, and A. Antonacopoulos, "The IMPACT dataset of historical document images," in *Workshop on Historical Image Processing*, 2013, pp. 123–130.

[9] W. F. Clocksin, "Handwritten Syriac character recognition using order structure invariance," in *Proc. 17th International Conference on Pattern Recognition*, vol. 2, Cambridge, UK, August 2004, pp. 562 – 565.

[10] E. Tse and J. Bigun, "A base-line character recognition for Syriac-Aramaic," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 1048–1055.

[11] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "Aletheia - an advanced document layout and text ground-truthing system for production environments," in *International Conference on Document Analysis and Recognition*, 2011, pp. 48–52.

[12] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "Competition on handwritten document image binarization (H-DIBCO 2014)," in *14th International Conference on Frontiers in Handwriting Recognition*, 2014, pp. 809–813.

[13] N. R. Howe, "Document binarization with automatic parameter tuning," *International Journal on Document Analysis and Recognition*, vol. 16, no. 3, pp. 247–258, 2012.

[14] N. Otsu, "A threshold selection method from graylevel histogram," *IEEE Trans. on System, Man, Cybernetics*, vol. 19, no. 1, pp. 62–66, January 1978.

[15] N. Howe, "Part-structured inkball models for one-shot handwritten word spotting," in *International Conference on Document Analysis and Recognition*, 2013.