Improving OCR for an Under-Resourced Script Using Unsupervised Word-Spotting

Adi Silberpfennig, Lior Wolf, Nachum Dershowitz
The Blavatnik School of Computer Science, Tel Aviv University
Tel Aviv, Israel
{adis2,wolf,nachum}@post.tau.ac.il

Seraogi Bhagesh, Bidyut B. Chaudhuri
Indian Statistical Institute
Kolkata, India
{to.bhagesh.sr,bbcisical}@gmail.com

***Abstract*—Optical character recognition (OCR) quality, especially for under-resourced scripts like Bangla, as well as for documents printed in old typefaces, is a major concern. An efficient and effective pipeline for OCR betterment is proposed here. The method is unsupervised. It employs a baseline OCR engine as a black box plus a dataset of unlabeled document images. That engine is applied to the images, followed by a visual encoding designed to support efficient word spotting. Given a new document to be analyzed, the black-box recognition engine is first applied. Then, for each result, word spotting is carried out within the dataset. The unreliable OCR outputs of the retrieved word spotting results are then considered. The word that is the centroid of the set of OCR words, measured by edit distance, is deemed a candidate reading.**

I. Introduction

Optical character recognition (OCR) for handwritten or old printed documents is notoriously difficult, making it nigh impossible to directly employ the obtained results without manual editing. We propose to alleviate this problem by using an efficient bootstrapping method for OCR betterment, based on the results of applying unreliable OCR to a dataset of unannotated documents.

The outline of the suggested process is as follows:

1) In the preparatory step, all images of a dataset A of documents undergo OCR by the baseline OCR engine. This results in a set of OCR results B. The bounding boxes of these results in the images are encoded by resizing each box to a fixed-size rectangle and then representing the outlined segment by conventional image descriptors. For accuracy and efficiency, a concise representation is extracted by performing maximum pooling over random groups of bounding boxes, using standard cosine similarity distances.
2) Given a new document requiring character recognition, the baseline OCR engine is first applied. Then, for each resultant word u, visual encoding is used, similar to that used for B. Word-spotting for u within the images A is then performed using a nearest-neighbor query on the set B. The set $C \subset B$ containing the n best word-spotting results is considered, each element of which has an associated textual reading provided by the baseline OCR.
3) Next, textual edit distances are computed between all pairs of words in the set $D = \{u\} \cup C$ of $n+1$ words, formed from the query word u along with the spotted words C. The candidate for improved OCR, v, is the centroid of the set D, i.e., the word with the least mean distance to the rest of the words in D. A score is assigned to the reading v based on the mean edit distance to the other elements in D and on the visual similarity between the bounding box of u and the bounding box of v.

The entire process is fully automatic and efficient. It uses only two parameters beyond those of the OCR and word-spotting engines themselves: the size n of the set of results that word-spotting returns, and a threshold θ used to decide whether to prefer v over u.

The next section briefly surveys some recent approaches to word spotting and OCR betterment. Then, in Section III, we explain the details of our method step by step, followed by a section devoted to the underlying OCR engine used. Section V presents experimental results on a dataset of Bangla documents. Bangla is a major South Asian script; the language, also called Bangla or Bengali, is used by about 270 million persons, mainly in Bangladesh and India. We conclude with a brief discussion of possible further improvements and extensions.

II. Background

Much effort has been devoted to research on word spotting; a few recent examples are [1], [2], [3], [4], [5]. Other works, with their own set of problems and less relevant here, deal with words embedded in outdoor photographs, e.g., [6].

Dynamic time warping (DTW) and hidden Markov models (HMMs) are two popular training techniques. An example of the former is [7] and of the latter is [8]. Many recent HMM-based systems are supervised and pre-segmented. Regrettably, these techniques are time consuming.

Two approaches to searching for occurrences of words in documents are possible: one can first segment the text images into words and then compare each target word with the query, or one can search for a match to the query using a sliding window of some sort. There is substantial literature on word segmentation, including, for example, [9]. An example of word spotting using segmented images is [10]; among the works that do not require segmentation are [11], [12], [13]. An in-between approach is to work with multiple overlapping target regions, as in [14]. Using multiple candidates for the purpose of reducing the number of false positives that sliding-window approaches can engender, is a current trend in computer vision; see [15], [16] among others.

Our word-spotting engine is based on the work of [17], which is inspired by the work of Liao et al. [18] in the domain of face recognition. Whereas [17] is an unsegmented word-spotting work, our method requires a tight coupling between the OCR and word-spotting results. Therefore, multiple adjustments are required. These include query jittering ([12], [18]) and a post-processing reranking stage.

Character recognition of printed text in Roman-based scripts is considered a solved problem since—for fair quality documents—OCR accuracy reaches 99.5% at word level. However, accuracy falls substantially when the document is of inferior quality, when it is old, or when it is printed in obsolete fonts. OCR engines for some oriental scripts are also quite advanced and have similar performance. However, the situation is not satisfactory for Indic scripts, for which the development of OCR engines is still at the laboratory stage.

There are several reasons for this situation. First, Indic scripts, such as Bangla, are alpha-syllabic, compared to Roman based scripts, which are alphabetic, with many fewer characters. Indic characters are divided into three categories, viz. basic, modified and compound; the number of distinct shapes that need to be recognized is about 1000. Character shapes have undergone changes over the past 200 years of printing, and orthography has also undergone modifications over this period, making a dictionary-based correction approach less effective. Furthermore, there is a resource crunch (of database and scientific information) for doing research in Indian languages and scripts that could otherwise be helpful for advanced OCR research. All the same, some good work has been done recently, and a workable OCR system for printed Bangla script has been developed lately (see Section IV). However, this system is not flexible enough to handle poor-quality Bangla text, and new approaches are required.

To the best of our knowledge, our work is the first to use word spotting in order to improve OCR results in an unsupervised manner. The work of Sankar et al. [19] is the closest work we are aware of. However, it deals with partial OCR, which is accurate where available, while we deal with noisy OCR.

III. Method Description

As outlined in Section I, our proposed method improves the OCR of a new document by considering each OCR result u and comparing it to a set of words B extracted automatically from a dataset of document images A. The method handles both the documents of dataset A and the query document in a uniform fashion, and the same processing pipeline is applied to both.

As a first stage, OCR is applied to each document image in A. The OCR results B include both the bounding box and proposed text of each recognized word. Word segmentation is therefore an integral part of the OCR engine. We do not make use of the quality score returned by the OCR engine, and we expect the OCR results to be noisy and only partly reliable.

Each OCR result is visually represented as a vector by considering the image patch of the associated bounding box, see Fig. 1. The patch is resized (by image interpolation) to a fixed size:160x64 pixels, where the parameters are obtained

Fig. 1. The patch normalization process. (a) Image patches obtained using the bounding box of the OCR engine. (b) Resized patches with grid overlayed. All bounding boxes of the set of documents and the target document are resized to the same size regardless of their size and aspect ratio.

from [17]. Using a regular grid, the fixed sized patch is divided into 20×8 non-overlapping cells of size 8×8, each of which is encoded by a HOG descriptor [20] of length 31 and by an LBP descriptor [21] of length 58. The HOG and LBP descriptors of all cells are concatenated and the resulting vector is normalized to a Euclidean norm of 1. The two descriptors form a single vector r of dimensionality $20 \times 8 \times (31 + 58) = 14,240$.

Since we rely on pre-segmented bounding boxes, which are not always exact, we apply a jittering process. Each bounding box is considered five times: the original bounding box, plus the bounding boxes that are obtained by shifting the original one 4 pixels in each of the four directions.

A matrix $M \in \mathbb{R}^{1000 \times 14240}$, which consists of the vector representations (same as r) for 1000 random OCR bounding boxes from the dataset B, is then considered. The vector r is transformed into a vector $s \in \mathbb{R}^{1000}$ by means of a linear projection $s = Mr$. In other words, the normalized descriptor vector is represented by its cosine similarities to a predetermined set of exemplars.

Then, a max-pooling step is carried out. The set of indices $[1..1000]$ is randomly split into fixed groups I_i of size 4. Given a vector s, this max-pooling is performed simply by considering one measurement per such quadruplet I_i that is the maximal value among the four indices of I_i in the vector v. Put differently, let t be the vector of length 250 that results from the max-pooling process as applied to the vector s. Then $t_i = \max_{j \in I_i} s_j$.

Given a new document requiring OCR, the black box OCR engine is applied to it. Then, each OCR result u is considered separately. The bounding box of u is resized as explained above, and the vector consisting of the multiple HOG and LBP descriptors is computed, multiplied by the matrix M, and max-pooling is employed using the same partition $\{I_i\}$.

This vector is then compared, by means of $L2$ distance, to the similar vectors—computed in exactly the same manner—of the B set of OCR results of the dataset A.

Note that the set of vectors associated with B is precomputed, which supports scalability. Since the representation is compact and the nearest neighbor search can be performed by means of matrix multiplication, the entire search process is performed very efficiently in main memory.

All elements of the set B are ranked in accordance with the computed $L2$ distance in $\mathbb{R}^{250}$. Unlike [17], we found the underlying encoding to be more reliable than the pooled similarity representation, and a reranking procedure is thus employed. The top $n_0 = 50$ query results are considered. For each, we compare the combined HOG+LBP encoding in $\mathbb{R}^{14240}$ by means of cosine distance to that of the word u. Then, the set C containing the top $n = 9$ results is considered.

The results in set C are fused to create a new OCR candidate v. We consider the set D of $n+1$ words that is the union of the word u with the words of C. Textual edit distances with a fixed and equal insert/delete cost is computed between $\binom{n+1}{2}$ pairs of words in D. The candidate for improved OCR, v, is the centroid of the set D, i.e., the word with the least mean distance to the rest of the words in D: $v = \arg\min_{w \in D} S_a(w)$, where $S_a(w) = \sum_{x \in D} d_{edit}(w, x)$.

The new candidate v is assigned a quality measure (lower is better) that is based on two factors. The first, $S_a(v)$ is the mean edit distance to the other elements in D . The second factor $S_b(u, v)$ is the visual similarity between the bounding box of u and the bounding box associated with v, as measured by the cosine distance of the joint HOG + LBP representation. The final combined quality score used is given by $S(v|u) = \log(e^{-S_a(v)/2} + S_b(u, v))$. The use of the exponent is done, as is often done, in order to covert a distance to a similarity.

Finally, the text of v is used in lieu of u for the same bounding box of u if $S(v|u) > \theta$. The default value of θ in our experiments is 0.25.

IV. Bangla OCR

An OCR system for Bangla has recently been developed at the Indian Statistical Institute, which works with about 98% accuracy for clean and recent documents containing text printed in the various fonts in modern character styles [22], [23]. The system begins with preprocessing, including noise removal and skew correction. This is followed by binarization, then text line, word and characters/subcharacter segmentation in the upper, middle and lower zones, after which the characters/subcharacters are submitted to a two-stage tree classifier. The first stage is a group classifier, wherein each group may consist of one or more similarly-shaped character classes. The groups are then subjected to second-stage classifiers to recognize the character/subcharacters of each group. This approach improves speed and offers flexibility in choosing different sets of features at the second level. Then the recognizer outputs of the upper, middle and lower zones are combined to form characters, and the characters are combined into words in machine code, with some simple post-processing—based on orthographic positioning rules—employed to correct a small amount of output errors. No dictionary or deeper linguistic information is utilized to improve results.

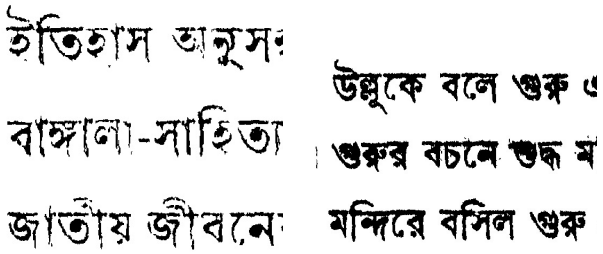


Fig. 2. Samples of the printed Bangla text. The printing is crude and the font is obsolete leading to poor OCR results.

The classifier is trained with the characters/subcharacters of a fairly large amount of text having ground-truth given in UTF8. Only texts in modern fonts (used during last 20 years) were employed in training the classifier. However, character shapes have undergone major changes over the past two centuries, during which characters in the Vidyasagari class of fonts, followed by Linotype and Monotype typefaces, slowly gave way to so-called "transparent" fonts developed with computer-based font generation software. Classifier training was mainly done using this last category of text.

While the results of this OCR system are fairly good on modern and clean documents, it is quite poor for old documents, especially those available on the net. We have tested the system on some pages downloaded from the website of the Digital library of India (www.dli.gov.in) in binarized form and obtained an accuracy of only about 54% at the word level. Some reasons for the poor results are that the documents are old and the quality of binarization is poor, while the font style is quite different from what this OCR system is trained on. Training with text in these old fonts is very difficult since there is little labeled data for infrequently used characters and no datum at all for the rare ones.

Overall, this OCR system is not versatile enough to recognize text from old documents. So, we set out to examine to what extent one can improve OCR accuracy by utilizing a word-spotting approach, without sacrificing efficiency.

V. Evaluation

Our method is evaluated on the Bangla dataset we downloaded from the Digital Library of India. The dataset contains printed text that was printed nearly 100 years ago using crude technology and an obsolete font. See Figure 2 for sample text. We used 18 pages comprising 3576 words that were manually annotated for evaluation purposes only.

For each query, we apply the jittering process, and all the elements in the set of the OCR results are ranked based on visual similarity. We then consider a set of 10 words, the top 9 results from the retrieved words and the query word. Out of this set a candidate is chosen to improve the OCR, see Figure 3 for examples. A quality score is then calculated for this candidate (see section III) and the text in the position of the query's bounding box is replaced with the candidates OCR if the score is higher than a threshold.

To evaluate the OCR betterment process we report OCR accuracy before and after implementing our system. We also evaluate independently the performance of the word spotting

TABLE I. OCR ACCURACY FOR THE BASELINE OCR METHOD, THE SUGGESTED PIPELINE, AND THE SAME PIPE LINE WHERE THE EDIT DISTANCE USED TO COMPARE OCR RESULTS IS REPLACED BY THE LONGEST COMMON SUBSEQUENCE SIMILARITY OR THE BAD OF LETTERS METHOD.

Method	OCR accuracy
Without implementing our method	52.0%
Complete pipeline using bag of letters	61.8%
Complete pipeline using LCS	64.2%
Complete pipeline using the edit distance	64.0%

TABLE II. WORD SPOTTING ACCURACY EVALUATED INDEPENDENTLY OF OCR IMPROVEMENT. WE PRESENT RESULTS FOR THE COMPLETE PIPELINE AND THE PIPELINE WITHOUT THE TWO IMPROVEMENTS INTRODUCED IN THIS PAPER.

Method	mAP
Complete pipeline	93.6%
Complete pipeline w/o query jittering	87.3%
Complete pipeline w/o re-ranking	89.7%
Baseline [17]	79.8%

system. For this, the mean Average Precision (mAP) retrieval score is used, according to reporting standards in the literature.

Table I shows the results achieved by our OCR improvement system and variants of it. We present results for our complete pipeline using the edit distance two alternative systems: one using a Longest Common Subsequence (LCS) based text similarity, and the second using a bag-of-letters representation. The length of the LCS is normalized by the length of the two words, which improves performance considerably. The bag-of-letters method simply represents each word by a histogram of letter frequencies, compared, as it gives best performance, by the cosine similarity. The LCS variant seems to perform slightly better than both alternatives and gives a sizable improvement of 12.2% in the OCR accuracy (over 23% relative improvement).

We studied the effect of two parameters on the system's performance: the size of the set of retrieved words n and the replacement threshold θ. The system is robust with respect to both. A value of n between 4 and 20 would give good result, and a value of $n = 5$ would give an overall OCR result of 65.2%. For the threshold, which is in log scale, once θ is below 0.20 all candidate replacements are made, above 0.30 none are made. Within the range $[0.21..0.28]$, the dependency between the threshold and the accuracy follows a smooth bell curve with a relatively large plateau between 0.24 and 0.26.

We also tried to apply the OCR improvement procedure iteratively, each time using the improved results from the previous round. The improvements were miniature: The second round contributed 6 more correct OCR results, the third round contributed one more correct word, and the process converged.

The performance of the word spotting method by itself, applied only on the Bangla pages which have ground truth is reported in Table II. Presented are results for the complete word spotting pipeline, and for the pipeline without the suggested modifications of query jittering and re-ranking. As can be seen, both help improve the overall word spotting quality.

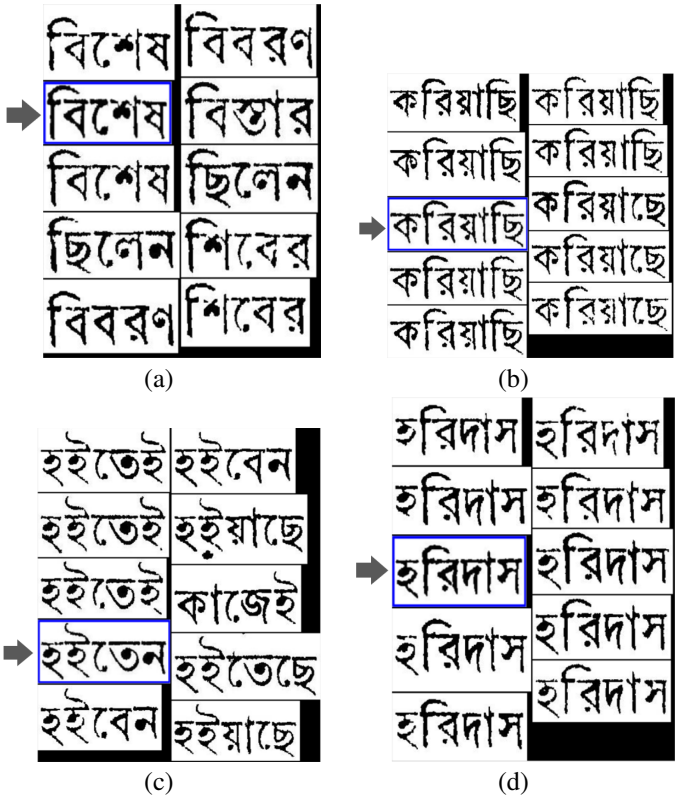


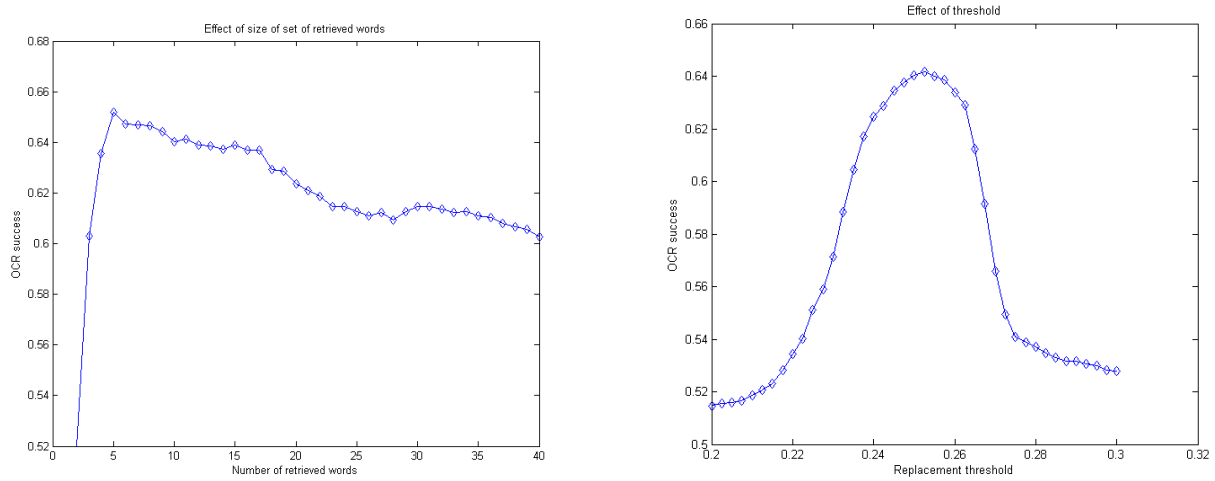
Fig. 3. Samples of OCR replacements, where the original OCR result was replaced by the new OCR candidate. In all cases, the original word is at the top left. The OCR of the marked words was the one selected for the replacement. (a,b) are good results. In (c) the word that was chosen is not the same as the query word. In (d) the word that was chosen is the same as the query word, however its OCR is incorrect. The figure could be misleading: there are many more occurrences of good replacements than of harmful replacements.

VI. CONCLUSION

Currently, as quality OCR technologies are still lacking when dealing with historical manuscripts, word-spotting technologies provide a useful substitute. In this work, we develop an unsupervised method for the improvement of OCR results that utilizes word-spotting. The method has the advantage that no ground truth OCR is employed during its application. Indeed, we use ground truth only for the purpose of experimental evaluation.

While it is possible to use ground truth combined with word spotting, in a similar manner, in order to obtain more accurate OCR results. This is of much less interest to the current effort, since it would lead to a fully supervised OCR method. The main advantage of the proposed method is that it can effectively utilize new collections and adapt to them, in order to improve OCR results, without any additional labeling effort. We are not aware of any other similarly unsupervised method.

The underlying word-spotting engine is extremely efficient. Indexing the words of each manuscript page is done in a few seconds, depending on the page's complexity. Retrieval requires a fraction of a second. The scalability and automatic nature of our method imply that it has the potential of becoming very useful in practice. It remains to be seen whether



(a) (b)

Fig. 4. The sensitivity of the proposed system to its parameters. (a) OCR accuracy vs. the size of the retrieved set D. (b) OCR accuracy vs. the score threshold θ used to decide whether to switch the OCR result to the new candidate.

OCR betterment can also be achieved on scripts with better developed OCR engines.

Lastly, the OCR hypothesis produced by the system is currently taken as the most central word (in edit-distance terms) among the automatic OCR of the retrieved words. It is sometimes the case that an out-of-sample word can have a lower mean edit distance. Such a word can be found using dynamic programming and later on may be verified using a dictionary. This is left for future work.

ACKNOWLEDGMENTS

The Tel Aviv University authors would like to thank the Ministry of Science, Technology and Space (Israel-Taiwan grant number 3-10341) and the German-Israeli Foundation for Scientific Research and Development (grant no. I-145-101.3-2013) for their support. This research was also supported in part by the Israel Science Foundation (grant no. 1330/14) and by the Friedberg Genizah Project.

REFERENCES

[1] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[2] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *ICDAR*, 2009.

[3] J. A. Rodríguez-Serrano and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," in *ICFHR*, 2008.

[4] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *PAMI*, 2011.

[5] K. P. Sankar, R. Manmatha, and C. V. Jawahar, "Large scale document image retrieval by automatic word annotation," *IJDAR*, 2014.

[6] K. Wang and S. Belongie, "Word spotting in the wild," in *ECCV*, 2010.

[7] T. M. Rath and R. Manmatha, "Features for word spotting in historical manuscripts," in *ICDAR*, 2003.

[8] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "HMM-based word spotting in handwritten documents using subword models," in *ICPR*, Aug 2010, pp. 3416–3419.

[9] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," in *Scale-Space Theories in Computer Vision*, 1999, pp. 22–33.

[10] J. Almazan, A. Gordo, A. Fornés, and E. Valveny, "Handwritten word spotting with corrected attributes," in *ICCV*, 2013.

[11] L. Rothacker, M. Rusiñol, and G. Fink, "Bag-of-features HMMs for segmentation-free word spotting in handwritten documents," in *ICDAR*, Aug 2013, pp. 1305–1309.

[12] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Efficient exemplar word spotting," in *British Machine Vision Conference*, 2012.

[13] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *ICDAR*, Sep. 2011.

[14] A. J. Newell and L. D. Griffin, "Multiscale histogram of oriented gradient descriptors for robust character recognition," in *ICDAR*. IEEE, 2011, pp. 1085–1089.

[15] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *CVPR*, 2006.

[16] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *ICCV*, 2011.

[17] A. Kovalchuk, L. Wolf, and N. Dershowitz, "A simple and fast word spotting method," in *ICFHR*, 2014.

[18] Q. Liao, J. Z. Leibo, Y. Mroueh, and T. Poggio, "Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines?" *CoRR*, vol. abs/1311.4082, 2013.

[19] K. P. Sankar, C. V. Jawahar, and R. Manmatha, "Nearest neighbor based collection OCR," in *IAPR International Workshop on Document Analysis Systems, DAS*, 2010.

[20] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *CVPR*, 2005, pp. 886–893.

[21] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *PAMI*, 2006.

[22] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system," *Pattern Recognition*, vol. 31, no. 5, p. 531, 1998.

[23] B. B. Chaudhuri, "On OCR of a printed Indian script," in *Digital Document Processing: Major directions and recent advances*, 2007.