Character N-Gram Spotting on Handwritten Documents using Weakly-Supervised Segmentation

Udit Roy, Naveen Sankaran, Pramod Sankar K.†, C. V. Jawahar
Center for Visual Information Technology, IIIT-Hyderabad, INDIA
† Xerox Research Center India, Bengaluru, INDIA

***Abstract*—In this paper, we present a solution towards building a retrieval system over handwritten document images that i) is recognition-free, ii) allows text-querying, iii) can retrieve at sub-word level, iv) can search for out-of-vocabulary words. Unlike previous approaches that operate at either character or word levels, we use character n-gram images (CNG-img) as the retrieval primitive.** CNG-**img are sequences of character segments, that are represented and matched in the image-space. The word-images are now treated as a bag-of-**CNG-**img, that can be indexed and matched in the feature space. This allows for recognition-free search (query-by-example), which can retrieve morphologically similar words that have matching sub-words. Further, to enable query-by-keyword, we build an automated scheme to generate labeled exemplars for characters and character n-grams, from unconstrained handwritten documents. We pose this problem as one of weakly-supervised learning, where character/n-gram labeling is obtained automatically from the word labels. The resulting retrieval system can answer queries from an unlimited vocabulary. The approach is demonstrated on the George Washington collection, results show major improvement in retrieval performance as compared to word-recognition and word-spotting methods.**

I. Introduction

The ability to build text-based retrieval systems over handwritten documents is very much an open problem, inspite of much success over printed documents. Handwritten documents present unique challenges such as cursive writing, varying styles across writers, and even from the same writer. As a result, segmentation and recognition of handwritten characters is difficult and unreliable. This is typically addressed in literature by two popular approaches that avoid explicit character segmentation/recognition: i) whole-word recognition and ii) word-spotting.

In word recognition, the word is over-segmented into frames or components, which are recognized individually and then jointly to produce a word-label. The underlying recognition mechanism typically uses a Hidden Markov Model (HMM) [1], or an Artificial Neural Network (ANN) [2], that outputs the most likely lexicon word for the given feature sequence. One of the main reasons HMMs and ANNs are popular for handwriting recognition is that they do not require character level labels to learn the models or segmented characters during testing. However, such stochastic models require large amounts of training data and expert design of the model structure. They also have a high computational cost during recognition.

On the other hand, *word-spotting* [3], [4], [5] uses a query-by-example (QBE) approach. The query image is matched across the word-images in the document images, similar to a CBIR system, and relevant images are retrieved. This approach circumvents the need for textual transcription of the document images. However, word-spotting approaches are not amenable to query-by-keyword (QBK), unless one uses expensive manual labeling [3].

Most holistic word based recognition and word-spotting schemes are limited to a constrained vocabulary that was seen during the training phase. This is a serious limitation to build a scalable retrieval system, since labeled data is expensive to obtain. Moreover, neither approaches can efficiently retrieve morphologically similar words, without using computationally intensive Dynamic Time Warping (DTW), sliding-window technique [6] or textual language models.

In this paper, we overcome the limitations of the approaches that operate either at component level or at word-level, by using the character n-gram image (CNG-img) as a retrieval primitive [7], [8]. CNG-images are formed as sequences of character segments from a given word-image. This formulation is quite different from text n-grams which are used to provide a statistical prior on character labels [9]. The CNG-img, on the other hand, are represented and matched entirely in the image space. With the CNG-img-spotting approach, we first build a QBE retrieval system over handwritten documents. Due to the representative capability of CNG-img, the system allows for retrieving morphologically similar words also.

Further, we extend the work towards building a text-based retrieval system (or QBK). To avoid explicit word-recognition, we instead convert the QBK into an exemplar image which can query the QBE system. While it is straightforward to identify exemplar images for in-vocabulary queries, word-level exemplars are unavailable for out-of-vocabulary (OOV) queries. However, the advantage of the CNG-img-spotting scheme, is that it suffices to obtain the exemplars for the CNGs in the query. In order to obtain labeled exemplars for CNGs, the word-images need to be accurately segmented and labeled, which could be challenging over cursive-written documents. We address this challenge by proposing a weakly-supervised scheme for CNG-img segmentation/annotation using labels given at the word-level.

While much of previous transcript-alignment work operated at the word-level [10], [11], there are a few recent works that explore automatic character level annotation given the word

labels [12], [13]. For example in [12], character labeling is speed-ed up using character clustering/retrieval, while in [13], a conditional random field is used to align the labels to the word-image components over Chinese/Japanese documents. However, the underlying matching occurs at character or component level. We believe that better segmentation accuracy could be obtained by matching at CNG level. The inference of CNG-img segmentation/labeling is performed using a framework similar to expectation-maximization (EM).

To summarize, the contributions of this work are:

1) A QBE retrieval system over handwritten documents, using the character n-gram image spotting framework, that can retrieve morphologically similar words.
2) A weakly-supervised learning scheme for CNG-img segmentation/annotation to obtain reliable exemplars.
3) A recognition-free QBK search system that uses a small set of labeled exemplars to enable retrieval over almost-unlimited vocabulary.

II. Handwritten Character n-gram Spotting

Character n-gram spotting was recently introduced to perform recognition-free retrieval over printed documents [7]. Unlike previous approaches that operated at either character level or word level, the CNG-img spotting approach uses sequences of character segments as the retrieval primitive. The character n-gram spotting framework begins with segmenting word-images to candidate character segments. All possible contiguous sequences of segments are considered as the CNG-images. The word-image is in-turn treated as a bag of its constituent CNG-images, which is a denser representation than characters, that also offers much flexibility in comparison of word images. Example CNG-img set for a given word image is shown in Figure 1.

There are many advantages of using CNG-img, instead of characters or words as a retrieval primitive. A CNG-img has more information than an isolated character, which enables improved matching with the same features. Also, a small training dataset could generate a large number of exemplars, given that each word of L characters emits $L \cdot (L+1)/2$ number of CNG-images. For reasonable size of n-grams, the number of unique CNGs is limited, hence allowing for easy indexing of the CNG-img associated with them. Further, when one considers the character as a 1-gram and the word as an L-gram, both character and word based approaches are subsumed within the CNG-img-spotting framework.

The process of CNG-img spotting can be summarized in three steps. Firstly, CNG-images are obtained from the document collection and represented in a suitable feature space. The features are indexed for quick retrieval. In the second step, given a QBE, the query is expanded into its constituent CNG-images. The features from the expanded query are looked up in the index of features, to obtain individual retrieval lists for each of the query-CNG-images. The final step consists of merging the retrieval lists appropriately to present the user with one ranked list of word-images.

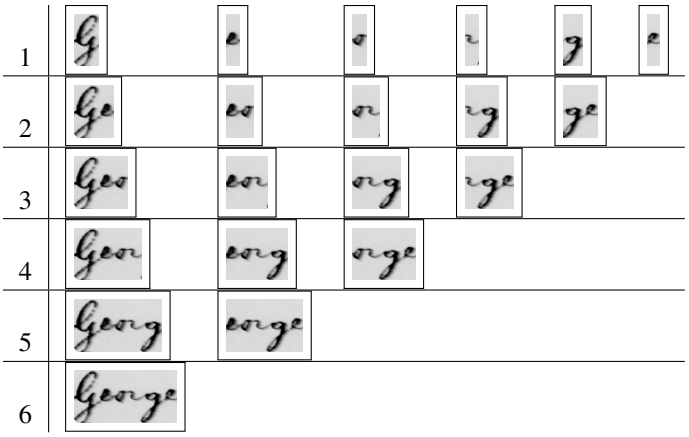

Fig. 1. An example word image and its corresponding character n-gram image set. It is important to note that we process character n-grams in the *image space only*, avoiding the need for explicit recognition.

By using a single index for all the n-gram levels, the approach robustly matches similar CNG-images inspite of degradations such as cuts and merges. Since the words are inherently represented at a sub-word level, CNG-img-spotting allows for easy matching of morphologically related words. Further, the spotting approach can be easily extended to QBK, by converting the query-keyword to a query-image by identifying a suitable exemplar image.

However, there are a few challenges with applying this framework to handwritten documents: i) segmentation of unconstrained handwriting is not easy, and ii) labeled exemplars are difficult to obtain at character/CNG level due to cursive writing. In the next Section, we shall address the problem of obtaining character/CNG level labeling of documents. The results from this are used to build a QBK retrieval system that can answer queries from an unlimited vocabulary set.

III. Weakly-Supervised CNG-img Segmentation

Given labels for word-images, the goal is to obtain character/CNG segmentation and annotation. In case of printed documents, the propagation of word to character labels can be performed using a simple connected-component analysis. However, character level labeling is not straightforward in cursive written documents.

We begin with three sets of data: i) weakly-annotated data that is labeled at word level, ii) strongly-annotated data of 300 words that is labeled at character level and iii) un-annotated data. Over the weakly-annotated data, the character segmentation is initialized using a weak feature, in our case we use character position in the word and the estimated character width. Let us denote the segment of character $c(i,j)$ of word W_i, by $\{L(i,j), R(i,j)\}$. The goal is to optimize the position of each $c(i,j)$ within the word, as well as ensure that the segment appears similar to other instances of the same character. This can be represented by the following objective function:

$$E = \sum_{i,j} \sum_{Ex_{c(i,j)}} dist(Ex_{c(i,j)}, \{L(i,j), R(i,j)\})$$

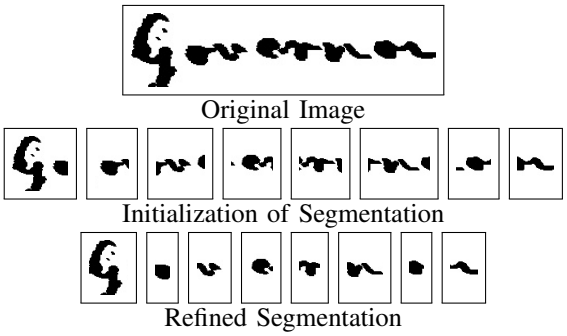

Fig. 2. Example of successful annotation refinement. The word "Governor" is segmented first into characters using a very weak feature (character width). The segmentation is then refined using strong features (Profiles) and matching technique (DTW).

where $dist$ is a function that computes dissimilarity between two image segments (or features); and $Ex_{c(i,j)}$ are exemplars for the character $c(i,j)$. In the CNG-img-spotting setting, $c(i,j)$ can represent the CNG-img in place of isolated characters. The exemplars $Ex_{c(i,j)}$ could be generated either from strongly annotated data (a case of weakly-supervised learning), or by the putative character segments from weakly annotated data (semi-supervised setting). In this paper, we restrict ourselves to the weakly supervised setting. The segmentation of the word-image into characters (and CNG-img) is the unknown parameter in this function. Optimizing the above objective function is typically performed using a two-step optimization algorithm.

In the first step, we assume that the segmentation of the CNG-images is provided and optimize the objective function on the appearance of the CNG-img segment against its expected appearance. The features from the segments are matched against those from exemplars in the strongly-annotated dataset. In cases where a CNG does not have an exemplar, it is generated by concatenating its corresponding character exemplars. The $dist$ function is the distance between the feature vectors of the character segment and the exemplar. We use the popular profile features [3] that consist of measuring at each column i) the number of ink pixels, ii) the number of background pixels between the word and the upper/lower word-boundary and iii) the number of ink-background transitions. Since the features are calculated at each column of the image, the feature length varies with the width of the image. These variable-length features are compared by finding the cost of Dynamic Time Warping (DTW) alignment.

In the second step, the appearance of the CNG-img segment is assumed fixed and the segmentation is optimized. This is performed using the alignment information provided by the DTW. Since we begin with a segment bigger than the actual n-gram, the backtrack of the DTW path will align the character sequences, while the beginning and the end of the segmented image would have a high 'insertion' cost [14]. The steps followed are: 1) compute the average cost of points in the backtrack path, 2) traverse from the beginning on the backtrack path and stop when the cost is below average; call this point estimated left-boundary, 3) perform similar traversal from the end of the backtrack path for the estimated right-boundary.

We propose two ways for refining the character segments given the estimates from DTW. The first method, called *nGramAvg.*, finds the character boundaries as an aggregate of the boundaries defined by all CNG that constitute the given character. For example, given the word "George", the left boundary of "o" is obtained as the aggregate of the left boundaries of "or", "org" and "orge"; similar procedure is followed for the right boundary. In the second method, called *nGramSub.*, the new estimate of character "o" is found by subtracting the surrounding n-gram boundaries from the word, i.e. $\text{"}o\text{"} = \text{"}George\text{"} - \text{"}Ge\text{"} - \text{"}rge\text{"}$. The same procedures are extended to refine CNG segments as well.

The new estimates for the segmentation is used in the next iteration of the algorithm, which is said to have converged when the segmentation estimates do not change beyond a certain empirical threshold. Example results from the segmentation/annotation procedure are shown in Figure 2.

IV. Text-querying with CNG-img Spotting

The segmentation procedure presented in the previous section generates labeled exemplars for a large set of CNG. Using these exemplars, one could use the QBE system described in Section II, to answer the text queries, hence enabling QBK retrieval.

Indexing Phase: The matching of CNG-images in the feature space is a computationally expensive task, since each word in the collection emits a large number of CNG-images. In order to speed-up the matching, we build an index using a combination of Hierarchical K-Means (HKM) and a random forest of KD-Trees [15]. To enable building an index, the features extracted from the CNG-img are ensured to be of the same dimensions.

Retrieval Phase: Using the built index, we obtain a list of approximate nearest neighbors for each Query-CNG-img. Thus, a candidate retrieval list is obtained for each unigram, bigram, etc. The task now, is to fuse the individual retrieval lists to obtain the final relevant image set for the given query. This is achieved by using a ranking function as described below. If Q is the query word with length L, then Q_i denotes the set of i-grams for the query. For each Q_i^j, the approximate NNs list is given as R_i^j. Each point P_k in the retrieved list R_i^j is weighted by its distance from the query as

$$S_i^j(P_k) = (2^{L-i} \cdot (L-i+1))^{-1} \cdot (1 - dist(P_k, Q_i^j))$$

The first term of the ranking function ensures that longer n-grams are given more weight than shorter n-grams. We choose to reduce the cumulative weight of each n-gram by half for each step of the n-gram. Thus, a K-gram will be given twice the weight of $(K-1)$-gram and so on. The second term is the distance of the retrieved CNG-img to the query CNG-img. The unique words from the retrieved lists of all query-CNG-img are scored by aggregating their corresponding $S_i^j(P_k)$

Algorithm	Annotation Error	Std. Deviation
Initialization	71.2%	59.2%
Characters Only	33.0%	36.9%
nGramAvg.	31.8%	33.7%
nGramSub.	**26.1%**	**29.7%**

TABLE I
PERFORMANCE OF SEGMENTATION REFINEMENT ALGORITHMS. THE NGRAMSUB METHOD OUTPERFORMS THE OTHER METHODS.

measure. The unique words are then re-ranked and presented to the user as the retrieval list for the given query image.

In-Vocabulary Queries: In the case of a text-query being present in the training data, the query is said to be "in-vocabulary". The exemplars for the query are obtained from the training dataset, which is used to query the QBE system. If multiple exemplars are present for the given query, better results could be obtained by using each of them as a separate QBE and aggregating the retrieved lists. Multiple exemplars are particularly useful while retrieving documents from different writers.

Out-of-Vocabulary Queries: Given a text query not seen in the training dataset, it is called an OOV query. The OOV query is first expanded to its n-grams in the text space (CNG-text). For each CNG-text, the training dataset is searched for the presence of an exemplar. If such an exemplar is present, it is used to query the index over the document collection to retrieve the approx-NN list. In cases where an exemplar is not present in the labeled dataset, an exemplar is synthetically created by concatenating exemplars of its constituent characters/n-grams. To speed up exemplar building, we pre-compute the features for the CNG-imgs and directly concatenate the features; the exemplar generation process only takes a few milli-seconds. The synthetic exemplars are now used to query the index. Due to the ability to construct any given query from its constituent CNG, the OOV querying mechanism can answer queries from an *unlimited vocabulary* set.

V. EXPERIMENTAL RESULTS

Experimental Setup: We evaluate our approach over the popular George Washington (GW) handwritten dataset [3]. The GW dataset consists of 20 pages containing more than 4700 words, written by a single author. We divide the dataset into two sets for training and testing, each contain 2300 words each. The training dataset is used to create labeled exemplars for the search system, using the segmentation approach presented in Section III. The testing dataset is used to evaluate the retrieval performance. The word-images are pre-processed to remove the slant from handwriting using a shear transform. We use the profile features [3] to represent the character n-gram images, which are known to be better suited for handwritten documents [3] and were shown to be robust to degradations [16]. Since profile features are dependent on word width, the images are scaled to a canonical size before feature extraction, to ensure uniform feature length while indexing.

Retrieval Scheme	Prec @ 10	mAP	Time/Query(sec.)
QBE Word Spotting (DTW)	0.52	0.49	15
QBE Word Spotting (L2)	0.29	0.21	0.24
QBE CNG-img Spotting	0.49	0.44	0.27
QBK In-Vocabulary	**0.61**	**0.57**	0.59
QBK Out-of-Vocabulary	0.24	0.18	0.59

TABLE II
RETRIEVAL PERFORMANCE OF THE PROPOSED SYSTEM, ACROSS VARIOUS QUERY AND ALGORITHM SETTINGS.

Segmentation Evaluation The segmentation error is defined as

$$\frac{SegmentWidth - Overlap}{GroundtruthWidth}$$

where $Overlap$ is the intersection between the segment and the groundtruth. The annotation refinement results are shown in Table I. The baseline method, given as **Characters Only** which uses isolated character segments refined by matching against character exemplars. We evaluate the error over character segments alone, in order to compare fairly with the baseline method. The two nGram based methods out-perform character based methods by a large margin. Among the two n-gram based re-estimation methods, *nGramSub.* performs slightly better than *nGramAvg.* The best performing setting has an error of 25%, which amounts to about 2 pixel error on either side of a typical 10 pixels width character. Much of the error is owing to the tight groundtruth segments, while the obtained segments contain some amount of cursive-connector pixels. Furthermore, the consistency of the segmentation is higher with *nGramSub* method, as evidenced by the smaller standard deviation of the error.

Retrieval Evaluation A few example retrieval results are presented in Figure 3. Given the query "Companies", our system was able to retrieve similar words such as "Company" in the Top-10 results. In case of the the query "receive", the erroneous result "inconceivable" is found due to the matching of the quad-gram "ceiv".

The retrieval performance of the CNG-img spotting framework is evaluated using two metrics: i) Precision among top-10 results, and ii) mean average precision (mAP). A retrieved result is said to match the query if their longest common subsequence (LCS), normalized by query length, is greater than a threshold of 0.5. The Average Precision (AP) is computed as the average of precision at each relevant retrieval for the given query. The mAP is the mean of the AP for multiple queries. It is essentially the area under the PR curve obtained from the retrieval evaluation.

The results are presented in Table II. The performance of our approach is compared against two word-spotting baselines, one that uses DTW to match the query with the collection and another that uses Euclidean distance. The DTW based word spotting performs well in the QBE setting, but takes close to 15 seconds per query using a thorough matching of the query across the collection. Faster DTW methods that use pruning [3] could be employed, but they typically result in loss of recall. The retrieval time is reduced to about 0.24 seconds by offline indexing using the $L2$-distance. It however, performs poorly compared to the proposed CNG-img spotting approach. In the

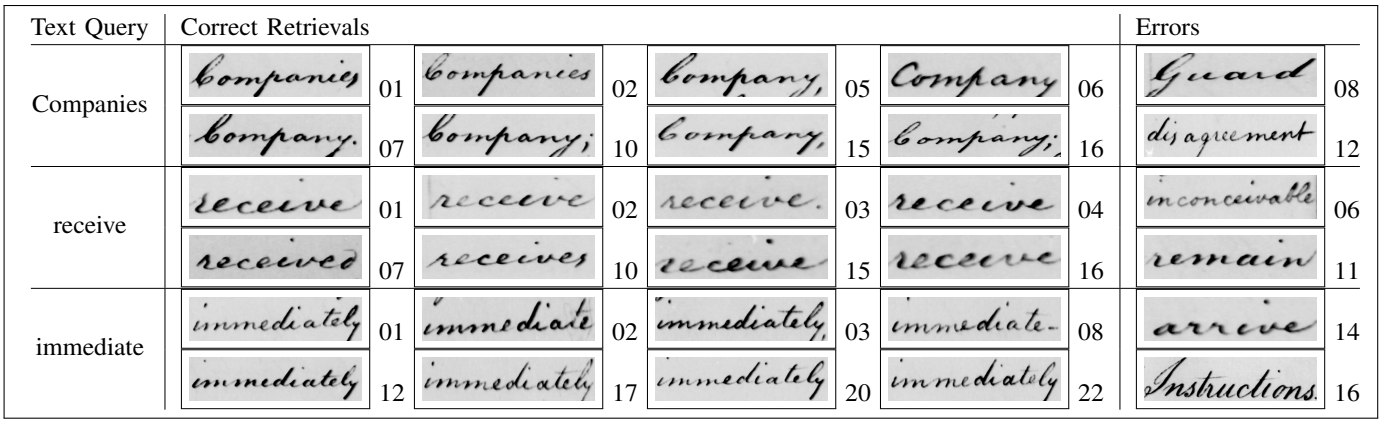

Fig. 3. Example retrieval results from our QBK retrieval system on the George Washington dataset. The results are obtained without explicit recognition or morphological analysis. As we can see the results are quite accurate, with similar words being retrieved automatically. A few errors in the retrieval are also presented.

QBK approach, we obtain a significant performance, given by a mean average precision of 0.5, for in-vocabulary queries. The performance drops for OOV queries, which is mostly due to similar CNG across words that are not morphologically related.

Time & Memory The proposed approach is also computationally efficient. The retrieval system uses 500 MB of RAM to index the feature set over the GW collection, which takes less than 440 seconds to build. Example query-time using different approaches is provided in Table II. All methods, with the obvious exception of DTW, have a sub-second retrieval time. During exemplar-building, the step of segmentation refinement takes about 1.1 seconds per word. The index size and indexing time scales linearly with the dataset, which means our framework is applicable to much larger collections. Further improvements could always be obtained by using more compact features or better indexing schemes.

VI. Conclusions & Future Directions

In this paper, we presented a scalable retrieval system over handwritten documents that allows: i) text-queries without explicit recognition, ii) sub-word retrieval without morphological analyzer and iii) search over unconstrained vocabularies without expensive computation. A comparison of our approach to the popular existing approaches such as word recognition and word-spotting is presented in Table III. It is clear that our approach overcomes many of their limitations. In future work, we would like to explore the robustness of the approach to multiple writers and multiple scripts. Most importantly, we shall use the CNG-img labeled exemplars generated from our weakly-supervised method to build a handwriting recognition system, using the method presented in [8].

Aspect	**Handwriting Recognition**	**Word Spotting (DTW based)**	CNG-**img Spotting**
Text querying	Yes	No	Yes
Retrieval time	Instantaneous	Time-consuming	Interactive
Degradation	Serious Effects on Segmn. & Recog.	Lesser Effects on Segmn. & Recog.	Segmn. errors don't matter
Data Scalability	Scalable	Not Scalable	Scalable
Vocabulary Coverage	Limited by Post-processing	Limited by Manual Annotation	Almost Unlimited
Morphological Word Retrieval	Requires Language Model	Requires Partial Matching	Inherently Addressed

TABLE III
A comparison of our approach with other popular approaches for handwriting retrieval. Our approach has multiple advantages over both approaches.

References

[1] Z. Lu, R. Schwartz, P. Natarajan, I. Bazzi, and J. Makhoul, "Advances in the BBN BYBLOS OCR system," in *Proc. ICDAR*, 1999, pp. 337–340.

[2] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. NIPS*, 2008, pp. 545–552.

[3] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, vol. 9, no. 2-4, pp. 139–152, 2007.

[4] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Proc. ICDAR*, 2011, pp. 63–67.

[5] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE PAMI*, vol. 34, no. 2, pp. 211 –224, 2012.

[6] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *Proc. ICDAR*, 2009, pp. 271–275.

[7] Sudha Praveen M., Pramod Sankar K. and C. V. Jawahar, "Character n-gram spotting in document images," in *Proc. ICDAR*, 2011, pp. 941–945.

[8] Shrey Dutta, Naveen Sankaran, Pramod Sankar K. and C.V. Jawahar, "Robust recognition of degraded documents using character n-grams," in *Proc. DAS*, 2012, pp. 130–134.

[9] S. Harding, W. B. Croft, and C. Weir, "Probabilistic retrieval of OCR degraded text using n-grams," in *Proc. ECDL*, 1997, pp. 345–359.

[10] C. Huang and S. N. Srihari, "Mapping transcripts to handwritten text," in *Proc. IWFHR*, 2006, pp. 15–20.

[11] E. Indermuhle, M. Liwicki, and H. Bunke, "Combining alignment results for historical handwritten document analysis," in *Proc. ICDAR*, 2009, pp. 1186–1190.

[12] J. Richarz, S. Vajda, and G. A. Fink, "Annotating handwritten characters with minimal human involvement in a semi-supervised learning strategy," in *Proc. ICFHR*, 2012, pp. 23–28.

[13] X. D. Zhou, F. Yin, D. H. Wang, Q. F. Wang, M. Nakagawa, and C. L. Liu, "Transcript mapping for handwritten text lines using conditional random fields," in *Proc. ICDAR*, 2011, pp. 58–62.

[14] M. Meshesha and C. V. Jawahar, "Matching word images for content-based retrieval from printed document images," *IJDAR*, vol. 11, no. 1, pp. 29–38, 2008.

[15] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. VISAPP*, 2009, pp. 331–340.

[16] Pramod Sankar K., C. V. Jawahar and R. Manmatha, "Nearest Neighbor based Collection OCR," in *Proc. DAS*, 2010, pp. 207–214.