Word Image Retrieval using Bag of Visual Words

Ravi Shekhar and C.V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India
Email: ravi.shekhar@research.iiit.ac.in, jawahar@iiit.ac.in

***Abstract*—This paper presents a Bag of Visual Words (BoVW) based approach to retrieve similar word images from a large database, efficiently and accurately. We show that a text retrieval system can be adapted to build a word image retrieval solution. This helps in achieving scalability. We demonstrate the method on more than 1 Million word images with a sub-second retrieval time. We validate the method on four Indian languages, and report a mean average precision of more than 0.75. We represent the word images as histogram of visual words present in the image. Visual words are quantized representation of local regions, and for this work, SIFT descriptors at interest points are used as feature vectors. To address the lack of spatial structure in the BoVW representation, we re-rank the retrieved list. This significantly improves the performance.**

***Keywords*-Word Image Retrieval, Bag of Visual Words, Scalability**

I. Introduction

Retrieval of relevant word images from a database of word images is a challenging problem. There are three primary dimensions to this problem: (i) How to represent the word images? (ii) How to match/compare two word image representations? and (iii) How to retrieve efficiently and accurately when the size of the database grows?. All these problems are relatively easy when the representation is text, which can be obtained using an Optical Character Recognition (OCR) system. However for many languages (especially for Indian Languages) reliable and robust OCR systems are still not available [1]. Many of these languages have rich heritage, and large quantity of printed material exist in them. They are now getting digitized and archived, but handicapped with the content level access to the collection [2].

Word spotting [3] has emerged as a promising method for recognition free retrieval. Here, word images are represented using some features, and comparison is done with the help of an appropriate distance metric. Due to appearance based nature of the matching, word spotting has the advantage that it does not require prior learning. Such word matching schemes have been popularly used in document image retrieval. For example, accessing historic handwritten manuscripts [3], searching documents in a collection of printed documents [4] etc. In traditional word spotting, word images are often represented using a sequence of feature vectors and compared using Dynamic Time Warping (DTW). Word spotting with DTW works well. However it takes approximately one second to compare two word images [3]. This makes it practically infeasible in case of large database, where millions of word images are present.

With the success of document image retrieval, scalability issues have surfaced. In [5], 10M pages are indexed, and the retrieval process takes only 38ms. This is achieved with the help of a memory intensive hashing scheme. The focus of their work is in retrieving *similar pages* with the help of an invariant descriptor. Such representations are too coarse for describing word images for the content level access. Methods like approximate nearest neighbor search [6] are also used to compare word images using a vector space representation. However such methods are also memory intensive. At the same time, we notice that text search engines are scalable to billions of documents comfortably. This motivates us to explore an alternative approach to word image retrieval.

We use BoVW representation for retrieval of word images. This is motivated by multiple factors (i) Bag of Words (BoW) representation has been the most popular representation for document (text) retrieval. There are scalable (and even distributed software) solutions available. (ii) BoVW method has shown to perform excellently for recognition and retrieval tasks in images and videos [7], [8]. (iii) Being a loose representation, BoVW representation can retrieve subwords, which is difficult with the popular vector space models. However, this paper does not exploit the full power of this flexibility in retrieving partial matches. In BoVW, an image is represented by an unordered set of nondistinctive discrete visual words. In retrieval phase, an image is retrieved by computing the histogram of visual word frequencies, and returning the word image, with the closest (measured by the cosine of the angles) histogram. This can also be used to rank the returned word images. A benefit of this approach is that, matches can be effectively computed. Therefore, images can be retrieved with no delay.

We argue that our method is highly language independent. The same visual vocabulary works well for multiple languages. We verify our method on more than 100K annotated word images in four different Indian languages. In order to demonstrate the scalability of the method further, we conduct experiment on a database of more than 1 Million words in Hindi. We measure the quantitative performance using mean Average Precision (mAP), and obtain an mAP of more than 0.75 across the entire collection.

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.96

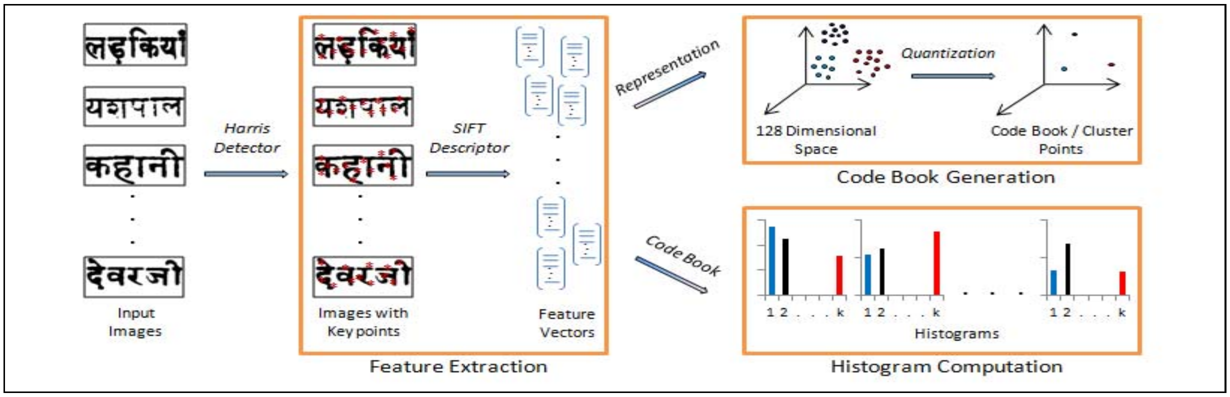

Figure 1. Bag of Visual Words Representation. Left: Input Images. Middle: Feature Extraction. Right Upper: Code Book Generation. Right Lower: Histogram Computation.

II. Bag of Visual Words

The BoVW model is inspired by the success of using BoW in text classification and retrieval. In BoW model, each document is represented by an unordered set of nondistinctive words present in the document, regardless of the grammar and word order. Document is formally represented with the help of frequency of occurrences (histogram) of the words in the vocabulary. These histograms are then used to perform document classification and retrieval. Analogously, an image is represented by an unordered set of nondistinctive discrete visual features. The set of these discrete visual features is called vocabulary. In the case of a document image, one can think of the glyphs as the vocabulary and a word can be defined as a bag of these glyphs. By representing an image as a histogram of visual words, one can obtain certain level of invariance to the spatial location of objects in the image. However, this creates certain issues in document image representation. For example, the word ‘DAS’ and ‘SAD’ are same for this representation due to the lack of order/structure in the representation. This reduces the precision in a retrieval task. We address this issue, while exploiting the computational advantages of the BoVW representation as explained in the next section.

Robustly segmenting a word image into the corresponding glyphs is practically impossible, specially for Indian languages where a single character (or connected component) can be composed of multiple glyphs. Moreover, in case of degraded documents, even extraction of characters become very difficult. Therefore, we represent the characters with the help of “interest points” like corners and blobs. At each of these interest points, we extract a Scale Invariant Feature Transform (SIFT) [9] descriptor to describe the local information as a vector of gradients. Space of SIFT descriptors is continuous, and we discretize the space by clustering SIFT vectors (often with K means) obtained from a small collection of documents. We use a vocabulary of 10000, and a clustering solution based on K(=1000) means is not scalable. We use a computationally efficient Hierarchical K Means [10] for this purpose. This algorithm clusters the data into C clusters first (where C is typically ≤ 10) and then samples in each of these clusters are clustered again recursively. This process is continued until we obtain the required number (in our case 10000) clusters. This can give more than 1000 times speedup in practice.

This visual vocabulary is then used to quantize the extracted features by simply assigning the label of the closest cluster centroid. This is carried out by rolling down the sample from the root to the leaf of the vocabulary tree [10]. The final representation for an image is the frequency counts or histogram of the quantized SIFT features $[f_1, f_2, ...f_i, ..., f_k]$ where f_i is the number of occurrences of i^{th} visual word in the image and k is the vocabulary size. To account for the difference in the number of interest points between images (due to size etc.), the BoVW histogram is normalized to have unit L1 norm.

Interest points are computed on word images using Harris corners. Harris corner detector is a popular interest point detector due to its strong invariance to rotation, scale and image noise. We also tried extracting the Maximally Stable Extremal Regions (MSER) [11] from the word images. However, that did not help much in our case. At each of the interest points, SIFT [9] descriptors were extracted. In SIFT, a neighborhood is described by a histogram of weighted gradients within a window to yield a 128 dimensional vector.

III. Retrieval System

In this section we describe our retrieval system. Figure 2 shows the overview of the system. The system is divided into two parts i.e., indexing and retrieval. This is in addition to the one time computation of the vocabulary. Indexing comprises of three steps as follows: (i) Features are extracted from word images, (ii) Histograms are created by vector quantization, and (iii) Database is created by indexing word images using an inverted file index. In retrieval process, first two steps are similar to the indexing. Then histogram is finally given to the index structure and images are retrieved in a ranked manner. We have used Lucene [12], a popular, reliable and open source search engine, for indexing.

Histogram computation (Figure 1) is carried out with the help of a precomputed vocabulary. For constructing the vocabulary, features are extracted from a subset of database images. Images are selected such that it covers all possible alphabets of the languages of interest. Clustering is done on the feature vectors extracted out of these images. Collection of centers obtained from this clustering is called the vocabulary as shown in Figure 1.

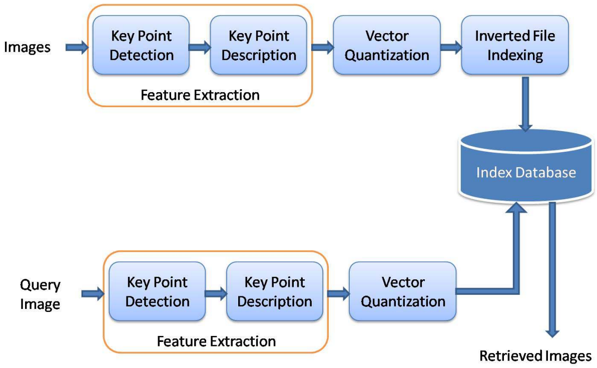

Figure 2. Overview of the indexing and retrieval.

A. Use of Search Engines

An inverted index is one of the popular and efficient indexing structures for BoW histograms. These index structures are implemented in many search engines. We use Lucene [12], a popular open source search engine for the present work. Each visual word (term) points to a list of word images (document) that contain it.

Internally, Lucene creates frequency file that contains the list of documents along with the term frequencies. If Lucene finds a term that matches with the search word in the term information file, it will visit the list in the frequency file to find which documents contain the term.

In retrieval phase, Lucene does a phrase (collection of terms) scoring. For a given phrase, approximate phrase *idf* is calculated with sum of terms. Then it calculates actual *tf* of phrase. Similarity between query and document is calculated by dot product of two histograms.

B. Enhancements

One of main limitations of BoVW is that it ignores the spatial relationships between visual words, i.e. it does not consider the order of the visual words. Therefore, the retrieved word images from the Lucene have poor precision. To overcome these limitations, we need to consider the order of the visual words. An elaborate storage of spatial information using a graph-like structure is computationally prohibitive. We use the the Spatial Pyramid Matching (SPM) [8] for this purpose. In this method, image is repeatedly subdivided and histograms of local features are computed at increasingly fine resolutions. In our case, we divide the image into three parts along columns only as shown in Figure 3. It was observed that, if image is divided into more than three parts, there is no significant improvement in the performance. The spatial order of the characters is thus enforced by considering the sub regions. Thus the first part of a query image can match only with the first part of a database image. Similarly for other parts.

C. Re-ranking

The initial retrieved results are then reranked explicitly to improve the overall performance of the system. Retrieved word images using BoVW and query word image are divided into three parts as explained earlier. SIFT matching is done for the corresponding parts i.e., original and three parts of both the query image and the top-k (in our implementation k = 250) retrieved images. We match the SIFT vectors by computing the distance between the SIFT vectors as well as the ratio of the best match to the second best match as in [9]. For two images $(I_1 \,\&\, I_2)$, score is given by the normalization of the number of unique match points with respect to the sum of number of features in both the images.

$$Score = \frac{\#Match\,Points}{\#SIFT\,in\,I_1 \;+ \#SIFT\,in\,I_2} \qquad (1)$$

Total score for a retrieved image is determined by weighted sum of scores of all parts.

$$Total\,Score = Score_{original} + \frac{1}{3}\sum_{i=1}^{3} Score_i \qquad (2)$$

where $Score_{original}$ is score for entire image and $Score_i$ is score for i^{th} part of the image. Retrieved images are reranked according to Total Score. Images with high score are kept at the top of the list.

IV. Results and Discussions

In this section, we present results to demonstrate the utility and scalability of the proposed system. To demonstrate the utility across languages, we use a large data set of 100K words. Datasets contain four different Indian languages (Hindi, Malayalam, Telugu and Bangla) with significant change in structure. Two of them have a headline and the other two do not have. Two of them Aryan languages and the other two are Dravidian languages. Details of the data set are given in Table II. All the books are annotated at the word level and ground truth was created using [13].

To evaluate the quantitative performance, multiple query images were generated. The query images are selected such that (i) They have multiple occurrences in the database, (ii) They are mostly functional words and (iii) They have

Table I
PERFORMANCE STATISTICS.

Language	#Images	#Query	Prec@10	mAP	Prec@10 after Re-ranking	mAP after Re-ranking	Prec@10 after Spatial Verification	mAP after Spatial Verification
Hindi	112677	138	0.8437	0.6808	0.8719	0.7820	0.8770	0.7865
Hindi	1008138	138	0.8059	0.5894	0.8509	0.7022	0.8543	0.7062
Malayalam	108767	101	0.7668	0.6962	0.8328	0.7991	0.8581	0.8188
Telugu	131156	131	0.8507	0.6483	0.8668	0.7328	0.8830	0.7495
Bangla	124584	125	0.8498	0.7806	0.9022	0.8766	0.9182	0.8947

Table II
BOOKS USED FOR THE EXPERIMENTS.

Languages	Dataset Type	#Books	#Pages	#Words
Hindi	Large	4	427	112677
Malayalam	Large	6	610	108767
Telugu	Large	5	742	131156
Bangla	Large	3	363	124584
Hindi	Huge	32	3992	1008138

no stop words. The performance is measured by precision at 10 (Prec@10) and mean Average Precision (mAP). The Prec@10 shows how accurate top 10 retrieved results are. Our method is giving 0.8543 Prec@10, even in the case of huge dataset (see Table I). The mAP is the mean of the area under the precision-recall curve for all the queries. A direct BoVW solution gave only a mAP of around 0.65. With our enhancements based on reranking and spatial verification, mAP increases to more than 0.75 as shown in Table I. (see columns 5 and 9). Some of the example queries and retrieved words are shown in Figure 4, where one can observe the print variations and degradations (like cuts and merges). It is also observed that, if the length of the query increases, the performance (mAP) also improves. This is shown in Figure 5. This is natural because, longer words have richer histogram and more discriminative power. In the case of shorter query words we were obtaining results where query is a substring of the retrieved word. We also analyzed the maximum possible mAP for the same retrieved list, that can be achieved with the help of an ideal re-ranking (i.e., all the correct images according to ground truth will be on the top of retrieved list). As it can be seen in Figure 5, our reranking method is quite comparable to the ideal re-ranking, especially for longer words.

To show the scalability, we use a huge dataset of 1M words in Hindi (see Table II). The retrieval time from Lucene required for this dataset is summarized in the Table III, on a system with 2 GB RAM and Intel® Core TM 2 Duo CPU with 2.93 GHz processor. Further the mAP for this dataset is comparable to that of huge dataset (see second and third rows of Table I). The drop in performance is of the order of 0.08, which can be attributed to the fact that the list retrieved by the Lucene is of the same size (in our size 250) in both the cases. It is natural that with huge dataset, we will have more occurrences (more than 250) in the database and a complete recall can not be obtained in the present setting. However, this can be easily improved by increasing the list from 250. Applicability of the method on a huge dataset verifies our claim that the proposed system is scalable to a huge dataset.

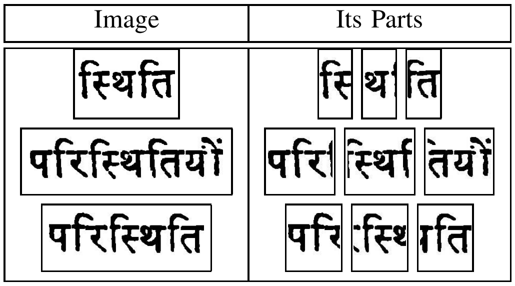

Figure 3. Spatial Verification.

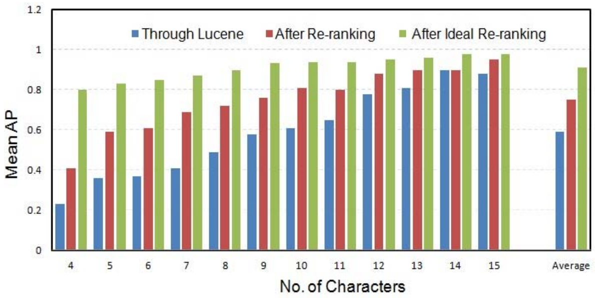

Figure 5. mAP Vs Query length. Also see the effect of re-ranking.

How good is our system?: To benchmark our results, we also considered a dataset of English words which are "visually" similar in quality. This is done by annotating English books from a public digital library. Using our method, we obtained an mAP of 0.77 for English. This validates

Table III
RETRIEVAL TIME.

#Images	Retrieval Time	Index Size
25K	50ms	28 MB
100K	209ms	130 MB
0.5M	411ms	550 MB
1M	700ms	1.2 GB

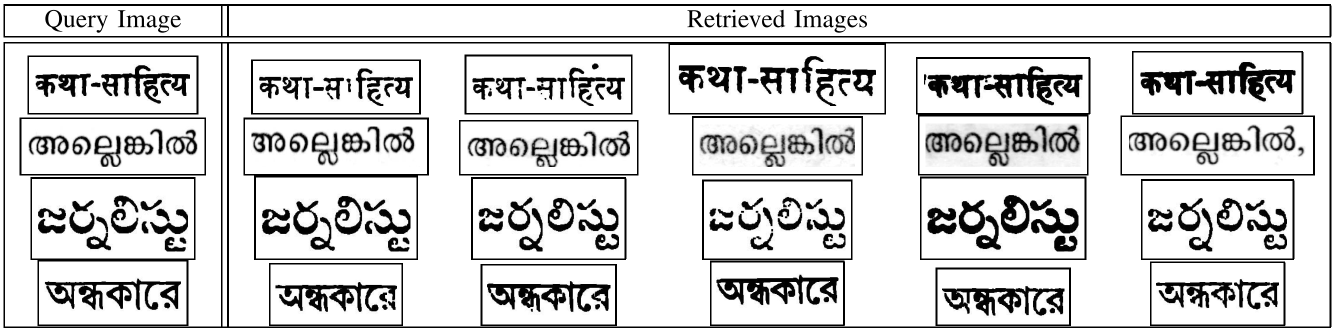

Figure 4. Results of Retrieval: First column shows the query images. Their retrieved images are shown in decreasing order, from left-to-right.

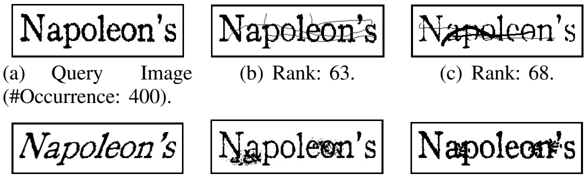
(a) Query Image (#Occurrence: 400). (b) Rank: 63. (c) Rank: 68. (d) Rank: 76. (e) Rank: 121. (f) Rank: 274.

Figure 6. Sample results for degraded Images, with rank.

that the results on Indian languages are quite comparable to those of English. To know the limits of degradation which our representation can handle, we created a set of degraded English words. Commercial OCRs failed to recognize these words. We indexed these words along with the dataset. We see that our retrieval system retrieves these words from the 100K database of English words. Some of the retrieved degraded words are shown in Figure 6 along with the rank. Note that there are 400 occurrences for this word in the database and the AP for this word is 0.7943. In general, we observe that our retrieval system is reasonably robust to cuts and merges in word images. SIFT descriptors are argued to be not robust to all possible document degradations. An improved descriptor can make our system compatible with a wide variety of document degradations.

V. Conclusions and Future Work

In this paper, we have presented a document retrieval system based on BoVW. Our method is highly language independent and scalable. The efficiency of proposed method is shown experimentally on four Indian languages. We have demonstrated the scalability of the method using 1 Million word images. Our future work includes (i) Learning document-specific local descriptors (ii) Use of better solutions than Lucene (iii) Use of noisy OCR outputs along with the BoVW representation (iv) Removing the re-ranking step, which is relatively time consuming.

Acknowledgements

This work is supported by Ministry of Communication and Information Technology, Government of India, New Delhi.

References

[1] S. Setlur and V. Govindaraju(editors), “Guide to OCR for indic scripts,” in *Springer*, 2009.

[2] K. P. Sankar, V. Ambati, L. Pratha, and C. V. Jawahar, “Digitizing a million books: Challenges for document analysis,” in *DAS*, 2006, pp. 425–436.

[3] T. M. Rath and R. Manmatha, “Word spotting for historical documents,” *IJDAR*, pp. 139–152, 2007.

[4] A. Balasubramanian, M. Meshesha, and C. V. Jawahar, “Retrieval from document image collections,” in *DAS*, 2006, pp. 1–12.

[5] K. Takeda, K. Kise, and M. Iwamura, “Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved llah,” in *ICDAR*, 2011.

[6] K. P. Sankar, C. V. Jawahar, and R. Manmatha, “Nearest neighbor based collection OCR.” in *DAS*, 2010, pp. 207–214.

[7] J. Sivic and A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *ICCV*, vol. 2, 2003, pp. 1470–1477.

[8] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *CVPR*, 2006, pp. 2169–2178.

[9] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[10] D. Nistr and H. Stewnius, “Scalable recognition with a vocabulary tree,” in *CVPR*, 2006, pp. 2161–2168.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *BMVC*, 2002.

[12] “Lucene,” http://lucene.apache.org/.

[13] C. V. Jawahar and A. Kumar, “Content-level annotation of large collection of printed document images,” in *ICDAR*, 2007, pp. 799–803.