TOWARDS EFFICIENT METHODS FOR WORD IMAGE RETRIEVAL

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science

by

Raman Jain
200502021
ramanjain@research.iiit.ac.in

International Institute of Information Technology
(Deemed to be University)
Hyderabad - 500 032, INDIA
July 2015

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Towards Efficient Methods for Word Image Retrieval" by Raman Jain, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. C. V. Jawahar

न हि ज्ञानेन सदृशं पवित्रमिह विद्यते।
तत्स्वयं योगसंसिद्धः कालेनात्मनि विन्दति॥

In this world there is no purifier as great as Knowledge; he who has attained purity of heart through prolonged practice of Karmayoga, automatically sees the light of Truth in the self in course of time.

Shrimad Bhagvad Gita (chapter 4, verse 38)

To my loving parents and siblings

Acknowledgements

First of all I would like to express my gratitude to Dr. C. V. Jawahar for his support, encouragement, motivation and for his inspiring supervision during the past four years. He has been a great influence on me as a researcher and has guided me in academic as well as non academic pursuits. I gratefully acknowledge all his help, patience, advice and effort to make me a better researcher and more accomplished individual.

I would also like to thank Dr. Anoop Namboodiri, Dr. Jayanthi Sivaswamy and Dr. P. J. Narayanan for their references and guidance in different subjects related to the stream.

My acknowledgement wouldn't be complete without the mention of my lab mates at CVIT especially Rahul, Omkar, Anand, Mrityunjay, Wasif, Mihir, Chandrika, Maneesh, Gopal, and Jinesh. I thank all my friends for their support and love.

Very special thanks go to my parents and my siblings, for their support, understanding, unconditional love and inspiration.

Abstract

The economic feasibility of maintaining large number of documents in digital image formats has created a tremendous demand for robust ways to access and manipulate the information these images contain. This requires research in the area of document image understanding, specifically in the area of document image recognition as well as document image retrieval. There have been many excellent attempts in building robust document analysis systems in industry, academia and research labs. One way to provide traditional database indexing and retrieval capabilities is to fully convert the document to an electronic representation which can be indexed automatically. Unfortunately, there are many factors which prohibit complete conversion including high cost, low document quality, and non-availability of OCRs for non-European languages.

Word spotting has been adopted and used by various researchers as a complementary technique to Optical Character Recognition for document analysis and retrieval. The various applications of word spotting include document indexing, image retrieval and information filtering. The important factors in word spotting techniques are pre-processing, selection and extraction of proper features and image matching algorithms. The Euclidean based algorithm is considered to be a faster matching algorithm. In the word spotting literature the Euclidean based algorithm has been used successfully to compare the features extracted from word images. However, the problem with this approach is that interpolation of images to get same width leads to loss of very useful informations. Dynamic Time Warping based algorithm is more preferable than Euclidean based algorithm.

In this thesis, a new approach based on Weighted Euclidean distance based algorithm has been used innovatively to compare two word images. The various features, i.e., projection profiles, word profiles and transitional features are extracted from the word images which are compared via Weighted Euclidean based algorithm with greater speed and higher accuracy. The experiments have been conducted on a large printed document databases of English language. The average precision rates achieved for this language were 95.48 %. The time taken for matching every two images was 0.03 milli-seconds.

A second line of research performed in this thesis considers keyword spotting for Hindi documents, the task of retrieving all instances of a given word or phrase from a collection of documents. During the course of this thesis, a novel learning based keyword spotting system using recurrent neural networks was developed. To the knowledge of the author, this is the first time that recurrent neural network based method is explored to Indian languages documents. In a set of experiments, its superior performance over state-of-the-art reference systems is shown.

Finally, the above system is applied for exhaustive recognition. The main findings of this thesis are that supervised learning in the form of training can increase the retrieval accuracy. The key to success lies in a well-balanced trade-off between data quality and data quantity when choosing the elements to be added to the training set. Performance evaluation using datasets from different languages shows the effectiveness of our approaches. Extension works are recommended that need further consideration in the future to further the state-of-the-art in document image retrieval.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Document image retrieval is becoming an attractive field of research with the continuous growth of interest in having information available in the digital format for effective access, reliable storage and long term preservation [22, 61]. Large number of digital libraries such as Universal Library (UL) [2], Digital Library of India (DLI) [1], and Google books are emerging for archival of multimedia documents. These documents cannot be stored as text always. This makes the search for relevant documents even more challenging. Nowadays, the storage devices are becoming cheaper and imaging devices are becoming increasingly popular. This motivates researchers to put efforts on developing efficient techniques to digitize and archive large quantity of multimedia data. The multimedia data includes text, audio, image and video. At this stage, most of the archived materials are printed books, and digital libraries are collection of document images. To be more precise, digitized content is stored as images corresponding to pages in books. These documents are typically available in very large numbers hence manually grouping and filing these documents for making them available easily is very tedious task. However, it is very important that these documents are made accessible to the users who would in fact like to search them with relative ease. Scanned or digital form of documents do not contain searchable text as it is, but contain words as images which cannot be searched or retrieved by existing search engines. Traditional text search is based on matching or comparison of textual description (say in ASCII/UNICODE) in association with a powerful language model. These techniques cannot be used to access content at the image level, where text is represented as pixels but not as text.

The best way of processing these digital documents is to segment the textual content present in the documents. Once the contents are separated out, a representational scheme (profile feature) can be applied to them to get their representational form, which could be ready to use in a content-based image retrieval system. Additionally, a direct method to access these documents is by converting document images to their textual form by recognizing text from images. Optical character recognizers (OCRs) are required to obtain the textual content from these documents [61, 8]. Success of OCR based text

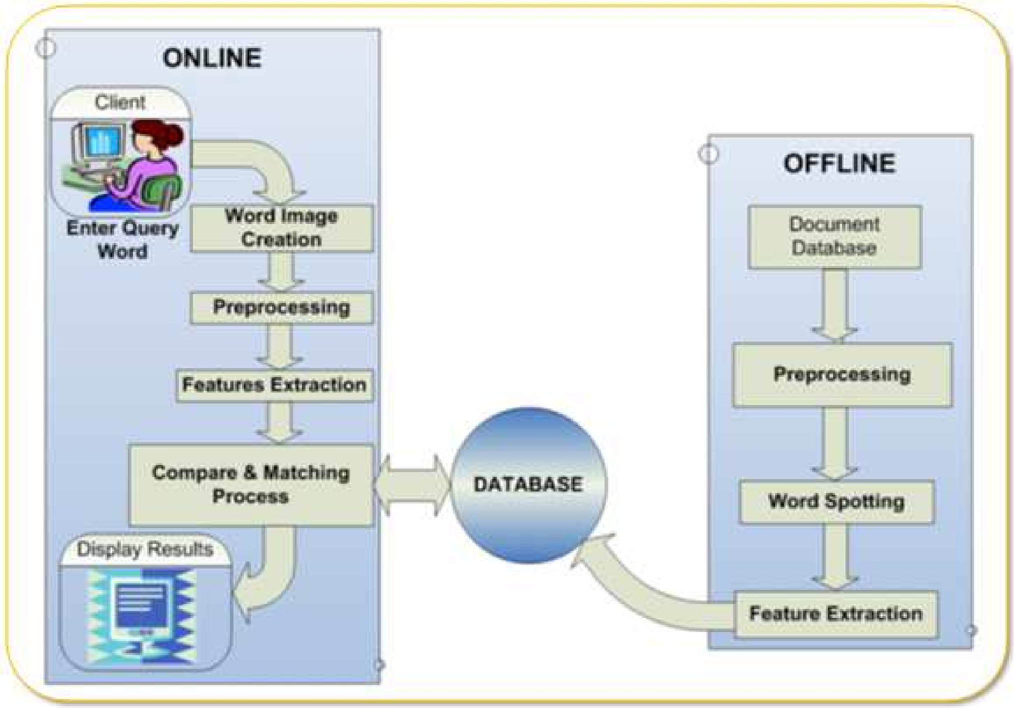

Figure 1.1 A simplified overview of document image retrieval technique

image retrieval schemes mainly depend on the performance of optical character recognition systems (OCRs) [75].

In literature, the use and application of OCR systems are well demonstrated for many languages in the world [61, 81]. For Latin scripts and some of the Oriental languages, high accurate recognition systems are commercially available for use. However, for Indian language with their own scripts, much attention has not been given for developing robust OCRs that successfully recognize diverse printed text images. Therefore, many alternate approaches are presented by researchers to access content of digital libraries in these languages [15]. The focus is on recognition free approaches for retrieval of relevant documents from large collections of document images. In this research work, we present new strategies and novel approaches that enable us in understanding and accessing the content of document image collections without explicit recognition. Our recognition-free retrieval scheme attempts to mimic the information retrieval technology in the image domain. A simplified overview of document image retrieval technique is shown in the Figure 1.1.

Handwritten or printed text along with pictures may also be present in these images. However, in this thesis, we limit our interest to printed document images like those from a book. Throughout the thesis, document is primarily used in this context.

1.1.1 Digital Library

A digital library [40] is library where enormous amount of documents are stored in digital formats which are searched or accessed using computers [31]. These documents could be in many forms such as handwritten documents, machine printed documents, or other media. The aim of digital library is to make these data available electronically on the Internet or on digital media. However, such large collections can only be accessed efficiently if a searchable or browsable index exists, just like in the back of a book.

There are a number of advantages of having Digital libraries over traditional libraries which include decreasing storage space, lowering in costs of library maintenance and resources, rapidly accessing books, archives and images of various types, and accessing widely distributed documents at lower costs.

A digital library is not only an automated conventional library, where the resources are electronically catalogued and are available only for browsing purposes, it also provides more equitable and widely distributed access at lower costs. Unlike traditional libraries, digital libraries have the potential to store much more information, simply because digital information requires very little physical space to contain them. Digital libraries can immediately adopt innovations in technology providing users with improvements in electronic and audio book technology. Also, it may be most appropriate means of organizing intellectual artifacts that cannot be represented or distributed in printed formats, such as audio/video multimedia content. Thus a digital library may evolve into a complex system that makes information available in hard copy, on magnetic tape and discs, CD-ROMs and videodiscs, including those from online sources. In summary, the following are some of the advantages of a digital library [3, 8].

- No physical boundary: The user doesn't need to go to the library physically. People from all over the world can access to the same information at any time, as long as an Internet connection is available.

- Information retrieval: The user is able to get access to any document by searching with any term such as word, phrase, title, name, subject etc. This is by the search engine to search in the entire collection that satisfy their need.

- Preservation and conservation: A number of copy of the original source can be made without any degradation in quality.

- Resource sharing: A particular digital library can provide a link to any other resources of other digital libraries very easily; thus a seamlessly integrated resource sharing can be achieved.

- Multiple accesses: Same source of digital library can be accessed by multiple users at the same time.

Large amount of archives of books and various collections of document images have been digitized. This enables the need for easy and rapid access. Indexing and retrieval are critical for success of any electronic library [49]. Effective access to such information sources is limited by the lack of efficient retrieval schemes. The use of text search methods requires efficient and robust OCRs. Hence, there is a need to explore search algorithms in the image domain.

1.1.2 Document Image Retrieval

In increasingly growing work in digitization environment, the document image retrieval techniques provide a solution to enable the archived valuable material searchable or accessible by users. This requires dedicated research in the area of document analysis and understanding, specifically in the area of document image retrieval. Searching and retrieval for relevant document images can be done with the help of the following approaches: Recognition-based and Recognition-free approaches. We are briefly discussing these methods in the following subsections.

1.1.2.1 Recognition based document retrieval

Recognition based approaches for accessing relevant documents requires an OCR system to convert document images into text contained in that particular image and then search in the recognized text. Effectiveness of recognition based approach greatly depends on the availability of robust character recognition systems. For most of the non-Latin scripts including indigenous scripts of Indian and African languages, the robust character recognizers are non-existent. Designing robust recognizers that are applicable for documents varying in quality, fonts, sizes and styles is known to be a long term solution. OCR based document retrieval accepts a scanned image of a text document. The image passes through many stages before it is translated into a computer understandable format. The image first undergoes some image enhancement for noise removal or increasing the contrast, then it goes for skew correction. Then, the skew corrected images are segmented to lines, words and finally characters in the text document, resulting in a whole bunch of smaller images, each representing a character or connected components. OCR is now used to recognize these smaller images. These recognized text can be made searchable using text based search engines.

1.1.2.2 Recognition free document retrieval

Recognition-free approaches search for retrieval of relevant documents without explicit recognition in the image domain. In contrast, document image retrieval without explicit recognition is a direct approach to access the digitized document image itself with certain level of performance. Specially, for historical printed and handwritten document images it is a promising approach to design an applicable retrieval system. Like recognition based approach, recognition free scheme also passes through many steps. Given a document image, it is preprocessed and segmented into word level. Then, features are extracted for representation of the segmented word images. For a given query, the same procedure of feature extraction is applied to it. The query is now compared with the images present in the database using a similarity function. Now, the images in the database will be ranked on the basis of similarity score. Such approaches can be made efficient using an indexing process.

1.2 Challenges in retrieval

Document image retrieval has many challenges due to the complexity present in the documents in terms of degradation, poor printing quality etc. Search and retrieval from the large collection of documents is the most important issue in document image processing field. Thus far, we have been considering only documents that were scanned in binary mode. Digitizing books in color would require more sophisticated scanners and better image processing and recognition algorithms and systems. If books are to be scanned in color mode, they have to go through a robust binarization system and then only those books can be retrieved. The documents, archived by digital libraries are not primarily of rich quality. The images contain variations in quality, script, font size and font style. Hence, retrieval task requires appropriate representational schemes and similarity measures, which can represent the content(text), while invariant to the popular variations. To design and implement a successful robust search engine in image domain, we need to address the following open issues.

- Unavailability of robust OCRs : The OCRs being used are not good enough for a powerful retrieval and search system. The current accuracy that they give is about 90% to 95% at character level and around 50% at word level. Hence, improvements in OCR technology are being awaited. Moreover, for Indian languages, OCRs are still in research phase, and satisfactory software system need to be built for OCRing Indian language content.

- Degradation: The degradation present in the documents make retrieval tasks more challenging and difficult. Documents in digital libraries are extremely poor in quality. The degradation can be present in the image in many ways such as:

  - Excessive dusty noise, logos, figures, printed and Handwritten text etc. In case of noisy images, the performance of OCRs degrades rapidly.
  - Due to Large ink-blobs, two characters or components can be joined together.
  - Cuts of arbitrary direction due to paper quality or foreign material
  - Vertical cuts due to folding of the paper
  - Due to the poor quality of paper and ink.

  We should take care of these artifacts present in the document when we are designing a retrieval engine.

- Speed : Due to enormous amount of document images, the normal retrieval system becomes slow in searching. The system has to go through many steps such as : binarization, skew correction, image segmentation, feature extraction, matching and retrieval of documents. In a typical book, there could be approximately 80,000 words hence any one of these steps could be computationally costly. Hence, these steps should be optimal individually. All computational operations should be done offline and do minimal operations during online retrieval.

- Cross-lingual retrieval : The documents that users need may be available in different languages. Most educated Indians can read more than one language. Hence, we need to design a mechanism that allows users to retrieve all documents related to their queries in any of the Indian languages.

- Non-European language : These languages (Indian, South African) pose many additional challenges. Some of them are

  - Lack of standard representation for the fonts and encoding.
  - Lack of support from operating system, browser, and keywords.
  - Lack of language processing routines.

  These issues add to the complexity of the design and implementation of a document image retrieval system.

1.3 Overview of this work

The thesis mainly aims at addressing the problems of effective and efficient retrieval in large document image collections with effective representation of the word images. The approaches proposed in the literature are not good enough in the presence of degradation in documents and are not scalable to large collections of documents. In the following sections we discuss the problem, contributions of this work and organization of the thesis.

1.3.1 Problem statement

The document image retrieval techniques available are not feasible for collections of document images. Most of the existing techniques based on complex image matching methods are not good enough for the large collection of document images. The major hurdles in achieving efficient search are the processing time and the word image matching techniques [71].

The search based on word image matching also contributes to the retrieval time. The word matching methods are expensive and become more expensive when words are represented by high dimensional feature vectors [70]. Hence, the problem is to search efficiently in a collection of word images represented using high-dimensional feature vectors.

A number of possible solutions are present in the literature. The goal is to retrieve the maximum number of relevant words from the collection. There is a need for high precision and recall along with the search speed. The representations for similar words may vary due to variations in image sizes, degradations like noise etc. The search technique has to take care of such variations also to achieve good performance. Therefore, there is a need to explore the characteristics of the representations and choose the right ones. We choose a group of well defining representations and examine their expressive power through a number of experimental results.

1.3.2 Contribution of this work

Retrieval is straight forward when the image collection is annotated. However, annotation itself is time consuming process. In this thesis, an efficient retrieval scheme for searching in large document image databases is presented. Some of our contributions are the following.

- Propose an efficient word image matching scheme based on fixed length representation to effectively compare printed document images in presence of script variability and word-form variations. Combined feature extraction scheme is also designed that is invariant to printing variations and degradation.

- We argue that query specific classification algorithm can be applied to improve the word image matching performance in a computationally efficient manner. The work also encourages to scale this method for large number of images. We show that the proposed method performs better than baseline system such as DTW and Euclidean based matching.

- Design an word retrieval solution for Devanagari printed documents in Indian language. The BLSTM neural network is used for training and testing. The images are preprocessed in order to have very few number of classes. We conducted a number of experiments to find out the performance of the method using different combination of features. The scalability of the technique is demonstrated on a collection of 6 books of Hindi language. The performance results are encouraging to extend it to other Indian languages. A BLSTM neural network is also used as recognition system for Devanagari documents. The language based issues have been analysed. Extensive amount of experiments has been done.

The thesis addresses the above problems in a general recognition free framework. However, the results of this thesis could be highly useful for developing recognition engines.

1.3.3 Organization of thesis

This thesis is decomposed into five chapters. The first chapter of this work presents a general introduction to the field of document image retrieval and recognition, with some information about digital library, and then the challenges faced in building a retrieval search engine. Motivation behind the present work and the major contributions are also briefly described.

A brief survey of the search and retrieval techniques available for document image collections is presented in Chapter 2. The different recognition based and recognition free retrieval methods are discussed and the state-of-the art is presented. The drawbacks of the methods and alternatives to be explored are also discussed briefly. Then we extend to general introduction to distance functions used for word image matching, and some reasons why distance function learning is an important and interesting learning scenario. We then provide a detailed state-of-art techniques used for learning distance functions.

An efficient technique for word image matching is presented in Chapter 3. The features used for representing word images are discussed in detail. The computational complexities of this technique and feasibility to the document image collections are discussed. Some applications of these learning distance functions in different domains are discussed to explore the possibility of usage in document images.

A new technique called BLSTM (Bi-directional short term memory) neural network is used for Hindi word image retrieval is discussed in Chapter 4. The procedure first removes the upper and lower zone from the word images and gives them to the training phase to train the neural network. The learned neural network applied on the test images and retrieved the relevant images from the word image collections for queries. The performance of the BLSTM based word retrieval is demonstrated with illustrations of results obtained from different experiments conducted. The technique to achieve scalability on large databases of document images is presented. The scalability and efficiency of the methods are proved with a number of experiments conducted on a collection of Hindi books. Also, the use of BLSTM neural network as a recognition system is presented for Hindi language documents.

Finally, Chapter 5 contains the summary of contributions of this work, possible directions of the work in future and final conclusions.

.

Chapter 2

Related Work

In this chapter, we present the literature survey of the document image retrieval and recognition techniques used for search in large image databases. A number of the various methods are reviewed here. We discuss some of the popularly used techniques, available in the literature, in text and image retrieval. Most of the image retrieval problems are modelled as similarity search. We look at the similarity search techniques and their applications to computer vision problems in this chapter. We then present the methods for learning a similarity functions for a specific problem. A brief review of the some standard similarity functions definitions is also provided.

2.1 Retrieval from document images

There have been a number of approaches proposed in recent years for efficient retrieval of relevant document images. Essentially, there are two types of techniques observed to search in document image collections. The first technique is to use optical character recognizers to convert the images into text and then apply a standard text based search engine. The second technique is to search directly in the images and ignoring the recognition task. In the following subsections we will briefly discuss about each of these techniques and the current research on this.

2.1.1 Recognition based document retrieval

Enormous amount of significant printed and handwritten documents are getting available electronically on the internet and stored as images in digital libraries as mentioned in Sec 1.1.1. The Digital Library of India (DLI) [1] and the Universal Library (UL) [2] are the main communities who have been doing this task and special mention should be given to them for their much required support. Hence, document images are considered as an information source. When the textual representation or the ground-truth is not known for these documents, search becomes practically impossible. Hence, fast access to the contents of the document images becomes an important and challenging problem in document image. In the first place this was done by manual intervention, some one has to indexed documents manually, which is an expensive and untrustworthy procedure. Then, optical character recognition (OCR) based approaches are being used. In this section we provide a comprehensive study on OCR based approaches.

In recognition based search and retrieval techniques, the document images are passed through an optical character recognizer (OCR) to obtain text documents. The text documents are then given to the standard text search engine to build the index which helps in accessing relevant documents very fast and efficiently. These methods perform well on simple non-degraded contemporary printed documents with a standard font. A nice survey of indexing and retrieval methods are available in [20]. In this, Doermann *et al.* presented an extensive research issues when the documents are retrieved using OCR based approach. The issues are more or less very much relevant to English language. OCRs do not perform well in the case of Indian and non-European language documents.

In these methods, word indexing techniques are completely rely on the text which is provided by the OCRs. In practice, for Indian languages or non-European languages, the OCRs give high error rate and thus incorrectly recognized words are also indexed. The recognizer makes similar mistakes for a particular word. If that word's frequency would be high and it appears in the index. In practice, a user gives the correct word as a query. When the index is searched, the actual word is never retrieved. There have been some strategies proposed to deal with OCR errors [82, 58]. In most of such methods, the uncorrected output given by OCR is used for word indexing and then query is compared with the indexed word with the help of string edit distance. Next, with the improvement in character recognition by OCR, Edwards *et al.* [23] proposed an scheme the convert Medieval Latin manusripts into text with generalized Hidden Markov Models (HMM) and then retrieval is done. They provide each character and their character n-grams for training and get transcription accuracy of 75%. To cope up with OCR accuracies, there is a work method proposed by Yasuto Ishitani *et al.* [37], that tolerates recognition errors by adopting keyword matching which searches for a string pattern from OCR results consisting of a set of possible character candidates. A search engine is built by Taghva *et al.* [82] for documents obtained after recognition of images. The search is based on the result of similarity between query word and the words which are present in the database. Another work has been published by Chan *et al* [13] for printed and handwritten Arabic documents that performs segmentation and recognition steps together. To perform letter discrimination, the method uses gHMMs with a bi-gram transition model and kernel principal component analysis (KPCA) / linear discriminant analysis (LDA). Similarly, the Byblos system [91] uses a approach to recognize documents where a line is first segmented out and then divided into image strips. Each line is then recognized using an HMM and a bi-gram letter transition model.

The success of the recognition based search and retrieval techniques depends on the performance of the OCRs [60]. If there is any indexing used after obtaining text from OCR, the indexing method should be very efficient. In the survey [20], Doerman reported that the performance deteriorates significantly when the character recognition error rate is greater than 5%. The survey provides methods developed by researchers to search documents without the need of complete and accurate conversion of their corresponding texts using OCRs. For Indian languages [81, 64] or for many non-European languages, the good recognizers are no-existent. There is an another survey available on earlier trends of document image retrieval in digital libraries [56]. Accuracy of the present OCRs for these languages falls below acceptable levels hence can't be used for searching or retrieval. These un-availabilities of OCRs call for a better approach when the task is to search in a documents.

2.1.2 Recognition free document retrieval

Recently, enormous amount of work has been published related to recognition free document retrieval. These methods are independent of OCRs accuracies and search directly in the images. A technique used alot in recent years for solving retrieval task is word spotting. Word spotting is a term that was taken from speech recognition field and introduced in document image processing. It is a method of finding and locating the given query word in a collection of document images. Here, document images are first segmented into word images and then treated as a collection of word images. Word Spotting technique in document image processing was first used by Manmatha*et al.* [55]. The method includes positioning of parts of text specified by query words, comparing a user given template of text image. It can be generalized to any document of any language using any alphabet or any representational form. Word spotting is not a simple task to implement since the matching procedure has to deal with many issues even for printed text like variations in fonts, variations in size of fonts, variations in the size of the words, printing quality, resolution etc. The task becomes more difficult in the case of historical documents because of issues like paper deterioration which includes noise into the images, printing quality that inserts uneven illumination etc. Many studies of word spotting have been published for both kind of printed [16] and handwritten documents [70, 38] in recent years. They are shown useful for locating similar occurrences of the query word. Also, these works vary in assumptions and settings on which they depend.

Rath and Manmatha [71] used dynamic time warping (DTW) based word-spotting algorithm to compute image similarities. In this work, word images are matched with each other and then clustered using k-means clustering algorithm. Each cluster is then annotated manually by a person. They have also shown that their approach produces much better results than a number of other techniques like shape context matching. Similar approach was also employed by Jawahar *et al.* [41, 9] where they use k-means or agglomerative clustering techniques for clustering the words for printed Indian language documents. The DTW, a dynamic programming technique popularly used to align sequences and estimates the similarity between the word images, where word images are represented as sequences of feature vectors, and the cost associated with the dynamic programming based alignment is considered as dissimilarity between them. The DTW based method was also used by Jain *et al.* [38] for indexing and retrieval of online documents. Similarly, Balasubramanian *et al.* [9] also performed word image matching for printed document retrieval. In this approach, they match whole word in printed documents using DTW based partial matching scheme, which was proved to be helpful in overcome the morphological difference between words.

Konidaris *et al.* [45] propose a method for keyword guided word spotting from historical documents. They have shown that instead of clustering word images and then manually annotating, one can provide text as query that can be used to synthesized image which is later used for matching. Finally, they use user feedback while retrieving to optimize the performance. Similarly, Jawahar *et al.* [41] also showed that in the case of printed books one can synthesize the query images from textual query for making the system more usable. Lu and Tan [54] propose an approach which is capable of searching a word portion in document images. A feature vector which depends on the character sequence in the word, is extracted for each word image. Then, an inexact string matching method is used to measure the similarity between the feature vectors. This method works for matching two document pages directly.

Chaudhury *et al.* [17] presented a method to query in printed documents at word level. They used typical structural characteristics of Indian scripts. Word images are represented in the form of geometric feature graphs and printed text is indexed using suffix trees. Another word-spotting based document retrieval is reported in [89], which addresses web document image retrieval problem by applying exact word matching procedure. In this work, word images are given as queries for searching and the retrieval task is done in two parts. In first part, the examined information of the documents is stored which is done offline. In second part, matching is done online. Nakai *et al.* [62] proposed a method of camera based document image retrieval which is characterized by indexing with geometric invariants and voting with hash tables. This method shows high accuracy with normal digital camera. Srihari *et al.* [80] proposed a document image retrieval using signature as queries. A signature retrieval strategy includes techniques for noise and printed text removal from signature images. In this method signature matching is based on a normalized correlation, similarity measure using global shape based binary feature vectors.

Kolcz *et al.* [42] propose a method for retrieving handwritten documents using a set of visual word image templates. They define a set of visual templates for keyword class of interest and execute a search for words exhibiting high shape similarity to the model set. Marini *et al.* [57] presented word-level indexing of printed documents. Their approach indexes homogeneous document collection by automatically adapting to documents that vary in scripts and font styles. Here, unsupervised character clustering is done using self organizing maps(SOM).

Ataer and Duygulu [7] have reported a method for retrieval from ottoman documents based on word image matching. The method first segments binarized document images into word images and then uses a hierarchical matching technique to find the relevant images to query word images. In the matching step, four distinctive features: word length, quantized vertical projection profiles and quantized vertical projection profiles of ascenders and descenders, are used for matching consecutively and after each test dissimilar images are discarded for the next test. A method proposed by Gatos *et al.* [27] uses word spotting technique for Greek typewritten manuscripts for which OCRs are non-existent. The method does not require any previous block or word segmentation step and based on block-based image descriptors that are used at a template matching process. To make it fast and efficient, they constrained the matching process only on certain regions of interest.

Unlike traditional OCR methods, word spotting is capable of taking advantage of the fact that within a corpus such as books collection the word images are likely to be similar to each other. Additionally, techniques like [13] that work with symbol level of the word images are very sensitive towards segmentation errors. For Indian languages segmentation at character level are very difficult due to the complex scripts that contain lots of dangling modifiers.

Success of word spotting techniques depends on the features used for the estimating similarity between the word images. Due to this, the extraction of suitable features for an application has been played as an important problem in pattern recognition [22]. Feature extraction helps to find an appropriate representational scheme for objects to be processed. A comprehensive review of the various feature extraction methods for document retrieval system is given in [22]. Features are the basis of content based image retrieval (CBIR) [74] and object classification [22], and it is a broad research area in the field of computer vision and pattern recognition.

More recently, Schomaker *et al.* [76] proposed a method for retrieval of handwritten lines of text in historical documents. The scheme uses a brute-force matching of line-strip images using a correlator and compared with feature based methods. Guangya Zhu *et al.* [94, 95] presented a signature-based document image retrieval system which automatically detects, segments, and matches signatures from document images with unconstrained layout and complex backgrounds.

Recently, Guangya Zhu *et al.* [93] proposed an automatic logo-based document image retrieval system. In this approach, they used logo detection, segmentation by boosting a cascade of classifiers across multiple image scales and logo matching using translation, scale and rotation invariants of shape descriptors. This method is segmentation free and layout independent. Ehtesham Hassan *et al.* [33] proposed a shape descriptor based document image indexing and symbol recognition. In this method hierarchical distance based hashing technique is used for document image indexing. Tilin Li *et al.* [50] proposed system for document image retrieval with local feature sequences. This method presents a fast, accurate and OCR-free image retrieval algorithm using local features and intrinsic, unique, page-layout free characteristics of document images.

2.2 Distance functions for Matching

As we can see from the above section that word spotting method becomes a vital solution to search a document if OCRs is not handy in use for document processing community. A holistic representation of word images have been used for similarity measurement and hence there have been always a similarity function needed for getting the similarity evaluated for two word images. Similarly, distance could be used for measuring how similar two words are with the help of distance function. A distance function is a function defined over pairs of datapoints. Given a pair of datapoints, the function produces a real (and possibly bounded) value, which measures the distance between the pair of points. We will briefly review some well known functions used for image matching in this section. While learning distance functions is a somewhat new area of research, lots of work has been done on this front to in recent years. A brief review of such distance functions will be presented in the subsection.

2.2.1 Standard distance functions

Let us now turn to a brief overview of several other distance (and similarity) measures which can be used for computing distance:

- Euclidean Distance - This distance measure is the most naive, well known and widely used distance function (which is also a metric). The Euclidean distance between two points x_i and x_j is defined by

$$D_{Euclidean}(x_i, x_j) = \sqrt{(x_i - x_j)^2} = \sqrt{\sum_k (x_{ik} - x_{jk})^2} \tag{2.1}$$

  This measure is used as a baseline measure for all the system.

- Manhattan Distance - The distance measures the shortest distance that one would be required to walk between the two points x_i and x_j if a city is laid out in square blocks. More formally, it is the sum of the lengths of the projections of the line segments between the points onto the coordinate axes of the coordinate system. This distance, originally proposed by Minkowsky is defined by

$$D_{Manhattan}(x_i, x_j) = \sum_k |x_{ik} - x_{jk}| \tag{2.2}$$

  Nikoleta *et al.* [21] suggested a word spotting method, which uses Manhattan distance as a distance measure to compute the distance between query word and the word present in the database. The proposed mehtod doesn't require training step. Word images are represented using global shape features, which are very general in order to capture as much as possible information about the word shape.

- Mahalanobis Distance - This distance measure, originally proposed by P.C. Mahalanobis (Mahalanobis, 1936) in the statistics community. It is based on correlations between different features within a feature vector. The Mahalanobis distance is a generalization of the Euclidean distance which also takes into account the correlations of the dataset and is scale-invariant. In its original form it measures the distance of any two vectors, based on the assumption that they originate from the same underlying distribution. Formally, given a distribution p which has a covariance matrix $\sum$, the Mahalanobis distance between two vectors x_i , x_j is given by:

$$D_{Mahalanobis}(x_i, x_j) = \sqrt{(x_i - x_j)^T {\sum}^{-1} (x_i - x_j)} = \sqrt{\sum_k \sum_l x_{ik} {\sum}_{lk}^{-1} x_{jl}} \tag{2.3}$$

  Note that if the covariance matrix $\sum$ is the identity matrix, then the Mahalanobis distance becomes the Euclidean distance. If we restrict the covariance matrix to be diagonal, we now have what is called a normalized Euclidean distance. While in its original form the Mahalanobis distance assumes that the datapoints originate from a probability distribution with a covariance matrix $\sum$. It can be shown that the distance is well defined for any positive semi-definite (PSD) matrix A. We will therefore denote a general Mahalanobis matrix by the symbol A. We will describe several algorithms for learning a Mahalanobis metric in next Section.

There are other distance functions too which have been used for word image matching. For matching binary images, Andreev and Nikolay [6] presented a work which uses a generalized form of Hausdraff distances. The method is used for Bulgarian typewritten documents. Hausdraff distance and its modification are good choice for measuring the word level similarities. Similarly, another work is reported in [53], which uses weighted hausdorff distance word matching. This distance facilitates the detection and location of the user-specified words in the document images and applied to English and Chinese language documents. A comprehensive study studies on distance functions used for word image matching can be found in [69].

2.2.2 DTW

DTW, dynamic time warping can be treated as a distance measure and uses dynamic programming based procedure to compute distance between two sequences. Euclidean distance can only be applied if the length of the sequences are same. If the sequences lengths are different, sequences first converted to same length using interpolation or extrapolation techniques which leads to lose of some important informations. Then, Euclidean distance can be computed. In such cases DTW based distance is a more suitable similarity measure than the Euclidean distance where sequences may be of different lengths and/or different sampling rates. To find the minimum distance between two sequences it maps each element of a sequence to multiple element of another sequence. The cost associated with dynamic programming based alignment is considered as dissimilarity between them and can be applied to compare two word images. Let the two word images (profile features) are represented as a sequence of vectors $X = x_1, x_2, ..., x_M$ and $Y = y_1, y_2, ..., y_N$. The DTW cost between these two word images is $D(M, N)$, which can be calculated in dynamic programming based approach using,

$$D(i, j) = min \begin{Bmatrix} D(i, j-1) \\ D(i, j) \\ D(i-1, j) \end{Bmatrix} + d(i, j) \tag{2.4}$$

where $d(i, j)$ is the cost in aligning x_i and y_j. DTW is used in several work such as [71, 70, 44] where it allows sequences to be stretched or compressed.

2.3 Learning distance functions

In a common paradigm for information retrieval, and specifically for image retrieval, distances between a query and items in the data base are computed, and the most similar items are returned. Hence, distance functions have been widely used in a number of application domain and various computational tasks. According to Aggarwal *et al.* [4], the process of designing application specific distance functions has intrigued researchers for over three decades. While there are several well known distance functions for datapoints that are represented in some vector space $\mathbb{R}^N$, there are many application domains in which the data instances are various objects which cannot be readily represented by a feature vector in some vector space. Classical examples are images, biological sequences, and text documents.

The importance of the distance function for learning algorithms has been gradually acknowledged in the last two decades years, and many distance learning techniques have been suggested for the improvement of classification, retrieval and clustering.

A wide variety of work has been done on this front in recent years. Distance learning algorithms basically either learn a distance functions or learn a representation transformation of the input data. In image retrieval, if we have a good representations of an image, means the features are representative and discriminate from the non similar class of images, then a simple distance function would be good enough to get high retrieval rate. If the representation we have, is not good, then we would need more sophisticated distance function in order to get good performance. Learning distance function algorithms are the best alternative when you can not rely on the representation techniques that is used.

There have been extensive work done in recent years where features are being learnt, usually called feature selection. A review of such methods can be found in [47]. Readers may also refer [32] for a more updated summary. Similarly, Feature weighting is a term used when one learns weights for the input features, which are then usually used in a KNN algorithm (see [87] for a review). While this is a richer input transformation than feature selection (it includes feature selection as a specific case), it is still very limited in terms of distance function learning, and can be equated with learning of a diagonal Mahalanobis metric.

In literature, research on learning Mahalanobis metrics is the most advanced, and enormous number of algorithms have been reported on it. While not optimal, learning a global Mahalanobis metric is simpler, and a more popular alternative. Lowe *et al.* [52] was the first to suggest a mahalanobis distance learning algorithm. They proposed a gradient descent based optimization technique to reduce the error, which is formulated using a stochastic formulation. By this method, they are able to learn a diagonal mahalanobis metric. In [67] a similar method is used to learn a full mahalanobis metric. Prior to this work, Xing *et al.* [88] suggested a method for learning a full mahalanobis metric. They were the first to suggest a method for learning full mahalanobis metric. The method attempts to find a Mahalanobis metric in which the distance between pairs of positively datapoints are as small as possible and the distance between negatively constraints pairs are larger than a threshold. This was done using gradient ascent algorithm. Some other related works are presented in [28, 79]. Recently, on the same work, Weinberger *et al.* [86] suggest the Large Margin Nearest Neighbour algorithm (LMNN), which learns a Mahalanobis metric with the help of semidefinite programming concepts. The algorithm ensures that the K-nearest neighbours of each data point always belong to the same class, while at the same time datapoints from different classes are separated by a large gap. The method doesn't depend on the distribution of the data and scales to larger number of classes.

Learning a distance function for KNN is also very popular and not radically different from metric learning for retrieval or classification. Improving a metric for NN classification using learning techniques are studied in [36, 34, 85]. Specifically, in KNN, only relations between close points count. Consider for example a metric in which a certain class is manifested in several distinct clusters, which are relatively well separated from other classes. This may be a good representation for KNN, but bad for clustering and instance based retrieval, since each instance is connected only to a small portion of the relevant class.

A recent study on this is presented in [66]. Here, authors present a taxonomy of distance functions in terms of how, where and when they estimate the metric and categorized distance function learning algorithms.

2.4 Discussion

Document image retrieval can be done with Recognition-based and Recognition-free approaches or a combination of them. We gave a brief comparison between recognition-based and recognition-free approaches. Recognition-based approaches needs an OCR system to convert document images into text and then search in the recognized text. Recognition-free approaches, on the other hand, search for retrieval of relevant documents without explicit recognition (in the image domain). Performance of recognition-based approach greatly depends on the availability of robust character recognition schemes. Unfortunately unlike English language, Indian languages do not have commercial OCRs. Recognition-free approaches are seen better alternative of OCR if retrieval is the requirement. Accordingly, the present recognition-free approach needs further consideration to come up with suitable classifiers for diverse document image collections. Also, these present methods are not efficient for search in high-dimensional feature representations of huge collection of document images. Therefore, a huge demand arises when both efficiency and robustness are the concern. In this thesis we investigate the recognition-free approach for retrieval of document image collections in Indian and English languages which is both efficient as well as robust. We have also gave a detailed study on various methods of learning a distance function, which can be used for comparing two word images.

Chapter 3

Towards Efficient Distance Function for Word Image Matching

Matching word images has many applications in document recognition and retrieval systems. Dynamic Time Warping (DTW) is popularly used to estimate the similarity between word images. Word images are represented as sequences of feature vectors, and the cost associated with dynamic programming based alignment is considered as the dissimilarity between them. However, such approaches are computationally costly when compared to fixed length matching schemes. In this chapter, we explore systematic methods for identifying appropriate distance metrics for a given database or language. This is achieved by learning query specific distance functions which can be computed online efficiently. We show that a weighted Euclidean distance can outperform DTW for matching word images. This class of distance functions are also ideal for scalability and large scale matching. Our results are validated with mean Average Precision (mAP) on a fully annotated data set of 160K word images. We then show that the learnt distance functions can even be extended to a new database to obtain accurate retrieval.

3.1 Introduction

Matching two word images by computing an appropriate similarity measure, has many applications in document analysis systems [73, 59, 9]. This includes applications in accessing historic handwritten manuscripts [72, 53], searching for relevant documents in a digital library of printed documents [9], holistic recognition [48] and enhancing OCR accuracies by post processing the classification results [68, 35]. In this chapter we aim at learning effective similarity measures, which are specific to word images. We limit our scope to matching printed word images. Though our approaches are demonstrated on English, our methods are language independent.

Though words can be matched by comparing holistic features [51], the popular approach for matching has been aligning sequences of feature vectors using Dynamic Time Warping (DTW) [71, 55]. A sliding window (or a vertical strip) is moved from left to right and features are computed for each window. This results in a sequence of feature vectors. For computing the distance between two such sequences, they are first aligned using dynamic programming. Cost of alignment is treated as the distance between the two sequences. To match sequences of length M and N, one needs to do $O(MN)$ operations. For retrieving from a large data set, query has to be matched with every word in the database. For large scale matching and retrieval, thus, this becomes the bottleneck. Replacing DTW by Euclidean distance (possibly at the cost of accuracy) has been a step towards scalability [46]. This can result in constant time (i.e., $O(d)$) retrieval.

Computing similarity based on a simple Euclidean distance does not use the *knowledge* that the comparisons are made between two *word* images. While computing the distance between two feature vectors (whether they are made out of sequences or holistic features), we need to note that the individual features need not be uncorrelated. Most of the comparison techniques also do not use the fact that the words are generated out of a language model. This could further impose constraints on the possible feature vectors which could be generated from words in a given language. In this chapter, we explore distance functions which can be learnt from examples. Our objective is to capitalize on the fact that not all possible combination of characters are valid in a language, and a distance function learnt from training examples can actually benefit matching and retrieval on a test (unseen or unannotated) data set. We validate our claims on annotated and unannotated data sets presented in Section 3.2. Since we have a reasonably large annotated corpus for comparison, we use mean average precision (area under the precision-recall curve) or mean precision for statistically validating the approach.

We start our experiments by comparing DTW and Euclidean distance in Section 3.3. We observe that it is not reasonable to conclude DTW is always superior to a fixed length representation. This gives us hope for building efficient, at the same time effective similarity measures.

We then show how a specific weighted Euclidean distance can perform superior in a given setting (say when the set of possible queries are known *apriori*). We design a query-specific classifier (QSC), which is obtained by learning the weights (parameter associated with the distance function) on a "training" data set (Section 3.4). However, this method is restrictive. We then extend QSC by systematically extrapolating the weights to get new (or unseen) query's weight. We demonstrate the performance of our method on a large corpus of more than five million words.

3.2 Datasets and Experimental Setting

We first summarize the experimental framework we use throughout the chapter. We consider three different types of data sets in English. They are aimed at quantitatively evaluating the performance of the matching methods, as well as demonstrating the generalization capabilities of learning schemes. Data sets can be summarized as:

- **Calibrated Data (CD):** To study the effect of font and size variations, we consider a calibrated data set of word images. They are generated by rendering the text and passing through a document degradation model [90]. Intensity of degradation is characterized using a scalar, and is used for computing the probability of degradation for the boundary pixels. We consider two subsets of this data set, CD1 and CD2. The set CD1 consists of 1000 words in multiple fonts and sizes. All images are equivalent to words typeset in 8pt to 15pt, and scanned at 300 dpi. CD2 is similar to CD1 but has higher amount of degradation.

- **Real Annotated Data (RD):** This data set consists of a set of words with their ground truth (text) associated to it. This data set is built out of 765 pages from scanned books, publicly available on the web [1]. All the words in these pages are manually annotated for conducting experiments and evaluating the performance. There are a total of 162,188 word images in this collection from four books which vary in fonts.

- **Unannotated Data (UD):** In addition to the completely annotated data sets mentioned above, we also consider a data set of 5,870,486 words which come out of scanned books. Since the data set is not ground truth-ed, it can be used primarily to evaluate the precision for selected queries and not the recall.

Examples from these data sets are shown in Figure 3.2. The examples demonstrate how the same word appears in all the three different data sets. Note that the words in all the three data sets are not the same. It depends on the content of the books. However, they overlap. We use these datasets for designing more effective similarity scores for the comparison of word images. We use these similarity score for retrieving similar word images. We evaluate the retrieval performance with measures such as (i) Precision, (ii) Recall, (iii) F-Score, (iv) AP, (v) mean of AP. Most of our results are presented using mean of AP.

Precision is the ratio of number of relevant images to the total number of images retrieved, for a particular query. It measures how well a system discards irrelevant results while retrieving. *Recall* is the ratio of number of relevant images retrieved to the total number of relevant images present in the database. It basically measures how well a system finds what the user wants. By changing a matching threshold, one can typically increase recall at the cost of precision. A precision-recall(PR) curve plots how the variation in one affects the other. *FScore* is the weighted harmonic mean of precision and recall and measured in isolation. It combines the precision and recall into one score. Average precision (AP) measures the area under the precision-recall curve. Average precision makes use of both recall and precision and encourages the relevant results to appear at higher ranked positions. Mean of the APs computed for multiple queries gives us *mean average precision (mAP)* [84]. We plot the precision recall graphs of two words (i.e., “even” and “think”) in Figure 3.1. (These are computed in RD as discussed later). It may be seen that the AP can be significantly different for different words. The area under the curve for “even” is 0.502 and that for “think” is 0.928. Also note that, with a given feature set, not all words are equally easy (or difficult) to retrieve. A probable conclusion based on this is that the matching scheme (or distance measure) we use need not be the same for all words.

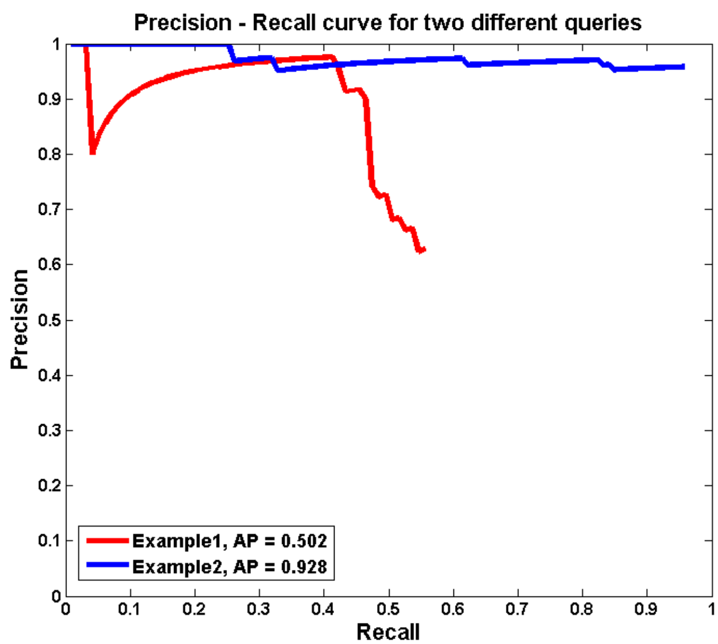

Figure 3.1 PR-curve on two different queries from the RD dataset.

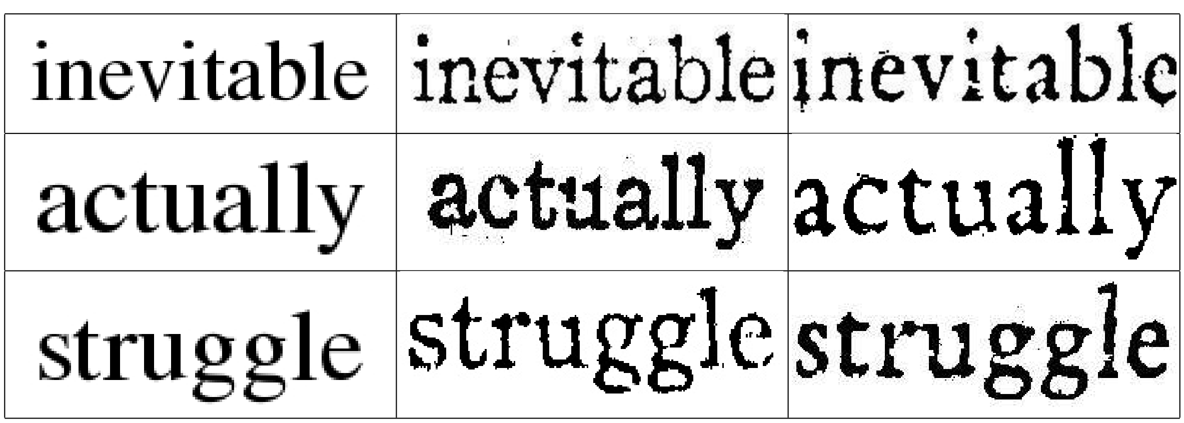

Figure 3.2 A comparison of words from three databases. Words in first column are from CD1, Words in second column are from RD and Words in third column are from UD.

Measure	DTW (w)					Euclidean (d)				
	w=1	w=2	w=3	w=4	w=5	d=80	d=90	d=100	d=110	d=120
Precision	0.653	0.512	0.432	0.320	0.271	0.594	0.596	0.596	0.598	0.599
Recall	0.805	0.793	0.787	0.731	0.731	0.793	0.793	0.793	0.792	0.787
FScore	0.721	0.622	0.558	0.445	0.395	0.679	0.680	0.680	0.681	0.680
AP	0.853	0.623	0.434	0.374	0.122	0.762	0.760	0.759	0.764	0.765

Table 3.1 Baseline results on comparing DTW and Euclidean distance on 300 queries from CD dataset.

3.2.1 Feature Extraction and Matching

We match and retrieve similar word images by comparing their feature descriptors. For the comparison purpose, we extract a set of profile features, which were also used in many of the previous works [71, 59, 44]. Thus word images are represented as a sequence of feature vectors. *Note that the objective of this work is not to propose any new set of features*, and we refrain from attempting that here. The features we use are (i) upper word profile (ii) lower word profile (iii) projection profile (iv) transition (black-white) profile. Feature extraction is summarized pictorially in Figure 3.3. More details can be seen in the previous works [71, 59, 44]. For each of the "query" word images, we compute the nearest K word images and compute the performance measures mentioned above.

All these profiles/features are computed on non-overlapping sliding windows (or a vertical strip) of w pixels. Prior to the feature computation, all blank vertical strips are removed. Since words can be of different length, the feature vector sequences can be of different length. This makes DTW a natural choice for comparison (and computing similarity). However, Euclidean and similar distances (L^P norms defined over a vector space) have many advantages in scalability and learning distance functions. We compute a fixed length representation (say of dimensions d) by varying w for each image.

Given a query image, we compute the distance of the query image from all the database images and find the nearest K matches. When query is text, an image is rendered first and the feature vectors are computed corresponding to the query word. Similar query processing was used in [9, 44]

3.3 DTW vs Fixed Length Matching

We first compare DTW and Euclidean distance based matching schemes. For the DTW, we vary w (in pixels, width of the non-overlapping window) and create feature sequences of varying length. For the Euclidean distance based matching, we vary the dimensionality — d (fixed length for all images) . We conduct our experimental study on CD1 and CD2 together. Results of this experiment are presented in Table 3.1. We compare the performance of DTW and Euclidean distance while computing the nearest K neighbors. For these experiments, K was set as 50. We observed that DTW gives superior results to Euclidean distance in all these cases. Possibly similar observations were also shown in some of the

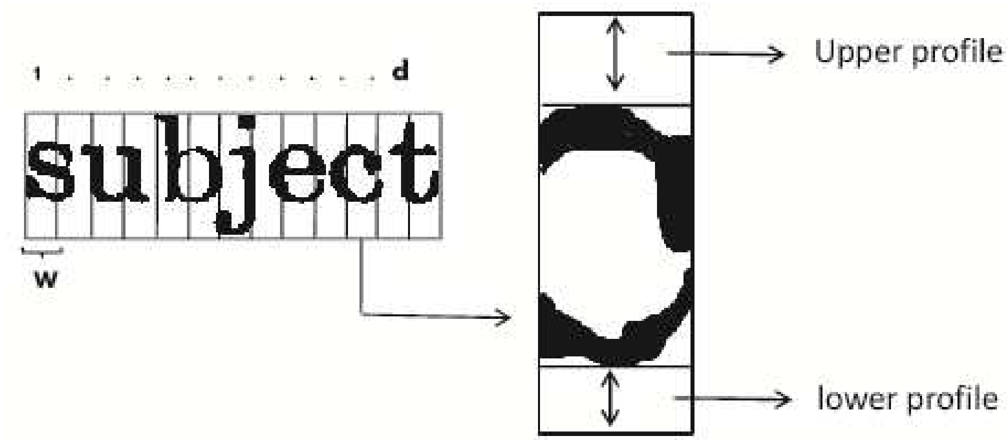

Figure 3.3 A pictorial representation of how profiles are computed and represented as a feature sequence.

previous works. (However, this is not always true as we show later in this section.) The success of this had been attributed to the fact that a dynamic warping can align well under the font/style variation of characters (either in print or in handwriting) and thus provide a better estimate of the dissimilarity.

However, it is reasonable to see that, when the number of fonts increases, both DTW and Euclidean distance based matching result in poorer performance. For example, if the mAP in a data set of 10 fonts in 0.829, the mAP in a data set of 50 fonts is only 0.453. We use our calibrated data sets for this study. This can be primarily attributed to the features/descriptors we use. However, this could be a general concern in many document image analysis tasks. (Interested readers may also see how character recognition can be poor when the number of fonts increase [18]). Figure 3.4 summarizes the effect of number of fonts on the retrieval.

We further conduct experiments to closely compare the performance of DTW and Euclidean distance based matching in a wider problem setting where data is real and having degradations. When the data is degraded, DTW is getting affected more severely than Euclidean distance. The results can be seen in Table 3.2. This can be partly attributed to the features and partly to the matching scheme. Figure 3.5 shows that the observation is consistent. i.e., with increase in degradation, DTW loses its advantages over Euclidean in the performance. In the rest of the chapter, we now investigate, how a fixed length representation can be matched better by learning an appropriate distance function. This is primarily motivated by the need to build a scalable, efficient and robust document image retrieval system.

Dataset	DTW	Euclidean
CD2	0.421	0.471
RD	0.778	0.817

Table 3.2 Results(mAP) on Degraded dataset CD2 and RD with 300 queries in each.

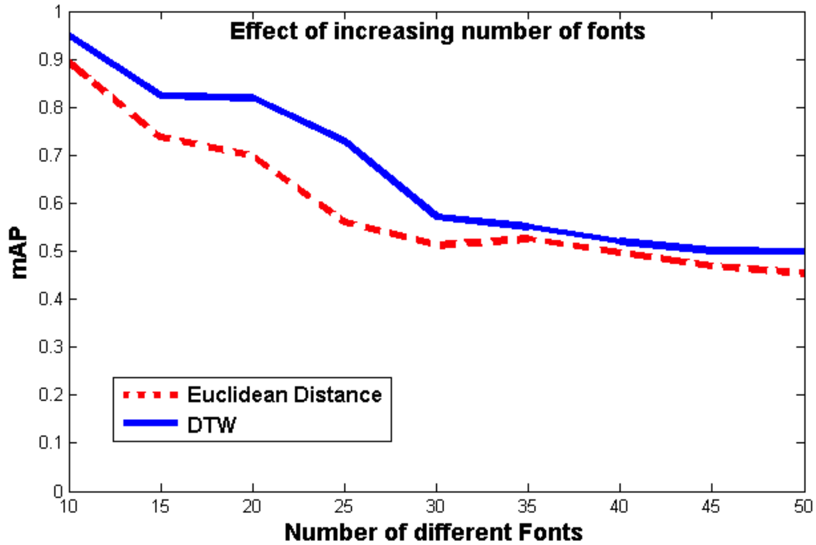

Figure 3.4 This graph shows how mAP changes when number of fonts increases.

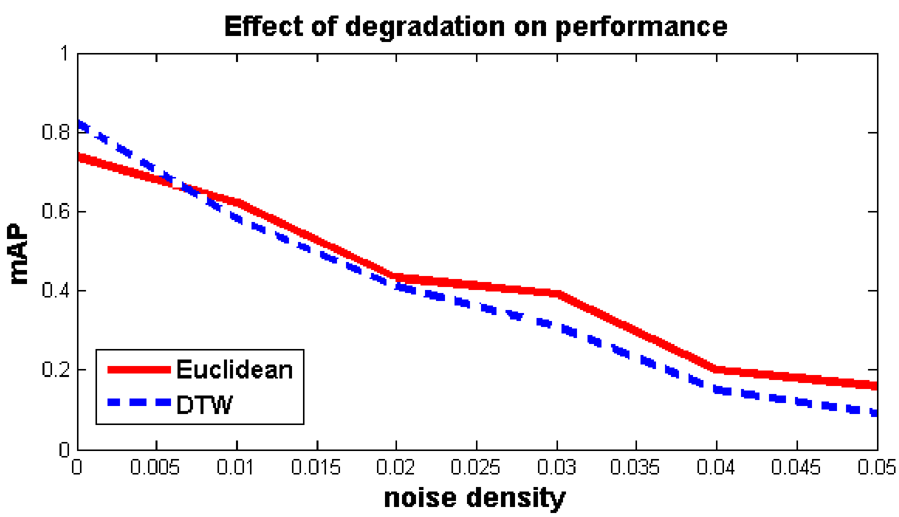

Figure 3.5 Performance comparison of DTW and Euclidean distance based matching with degradations in data

DTW is computationally intensive compared to Euclidean distance as shown in Figure 3.6. For example, DTW takes approximately 16 secs while Euclidean distance takes only 0.3 secs to find the K nearest match in a database of 162,188 words. To match sequences of length M and N, DTW needs to do $O(MN)$ operations while using Euclidean distance, one needs to do $O(d)$ operations. For the comparison, we set the dimensionality for the Euclidean distance matching as the average of the sequence lengths in the database (in our case 110) and the algorithm have been implemented in C++ and run on a 2Ghz Dual core machine running Linux. It is well known that the fixed length distance based matching scheme is scalable to large collection. DTW based matching is thus not suitable for

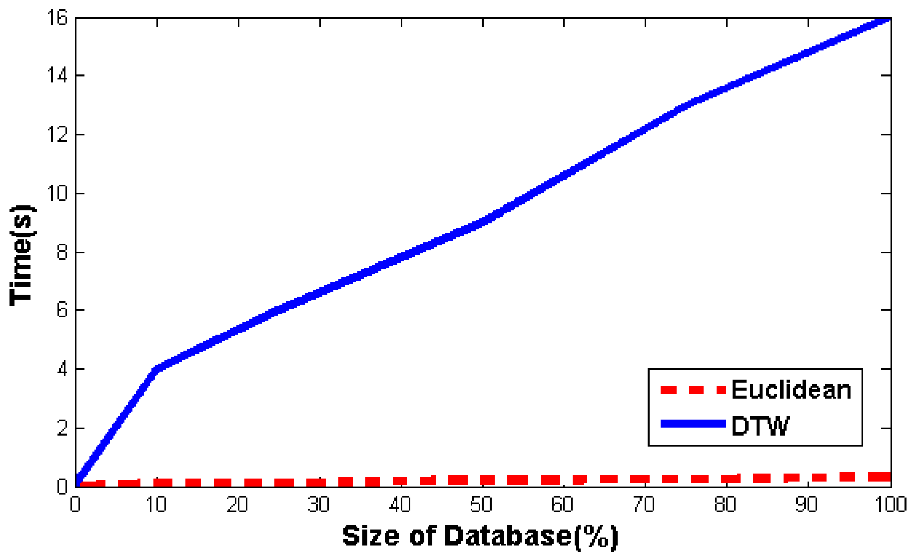

Figure 3.6 This graph compares average time taken by two different methods, DTW and Euclidean Distance, per query on increasing database size.

such an indexing. There are many useful data structures and indexing schemes proposed in literature to address this problem. Interested readers may find specific details at [12, 20].

3.4 Retrieval by learning query specific classifier

We are interested in the following problem: given a query word image, we would like to retrieve all similar word images from a database of word images, in a ranked manner. This retrieval problem can be understood as the design of an appropriate classifier for a given query. If one considers a nearest neighbour classifier, then the retrieved set is the ranked list of word images sorted in the increasing order of distance $d(\mathbf{f_i}, \mathbf{q})$, where $\mathbf{q}$ is the query word image and f_is are the database images. In the simplest form, this could be an Euclidean distance $\sqrt{\sum_j (f_i^j - q^j)^2}$. Where q^j is the jth feature of the feature vector representation. However, this may not be the ideal manner in which we can design the classifier. This is because: (i) this does not consider the information that f_i is in fact a fixed-length-representation built out of a sequence representation (ii) there exists a latent language model which generates these sequences and assign different probabilities for generation of words (iii) the classifier (or distance function) used to rank the word images could be different for each query. This necessitates exploring for a query specific classification scheme. Mahalanobis distance could have taken care of some of the aspects related to correlation of feature vectors.

We use a weighted Euclidean distance for matching word images and retrieving relevant documents

$$d'(\mathbf{f_i}, \mathbf{q}, \mathbf{w}) = \sqrt{\sum_j w^j (f_i^j - q^j)^2}$$

No learning					Learning with + examples					Learning with + and - examples				
French	French	French	French	France.	French	French	French	French	French	French	French	French	French	French
French	French	France.	France.	France.	French	France.	French	France.	France,	French	France.	French	French	French
after	after	after	after	offer	after	after	after	after	after	after	after	after	after	after
offer	after	after	offer	alter	offer	after	offer	after	after	after	after	after	after	alter
should	should	should	could	should	should	should	should	should	should	should	should	should	should	should
should	should	should	could	should	should	should	should	could	could	should	should	should	should	should

Figure 3.7 Example of images retrieved with and without learning, shown according to the rank in row major

where $\mathbf{w} = [w^j]^T$ is a weight vector. We learn $\mathbf{w}$ from examples. Such learning per query has been used in relevant feedback literature for retrieving documents [78] as well as images [19, 5, 24]. In general, in these two cases, the weight vector is learnt using the user feedback (User labels each of the retrieved results as positive or negative. Thus the user refines the retrieved results by continuous dialogue). During retrieval, in each of the iterations t, weight is typically refined (with any of the below equations) as:

$$w^j(t+1) \leftarrow w^j(t) + \frac{1}{\sigma_+^j} \tag{3.1}$$

$$w^j(t+1) \leftarrow w^j(t) + \frac{\sigma_-^j}{\sigma_+^j} \tag{3.2}$$

where σ_+^j is the standard deviation of the jth feature of positive examples. Similarly σ_-^j is for negative examples. In the first case, weights are learnt only from the positive examples, while in the second, weights get refined based on positive and negative examples. Both these (and variants of these) are popular in relevant feedback literature [92]. However, our setting is some what different from the relevance feedback methods. We are not looking for users to *interactively* retrieve word images. We are interested in learning a query specific classification/ranking scheme from a set of training examples, and directly use them on the test or new data.

Learning using Equation 3.2 is mildly different from a normal setting in our case. We partition database images as positive, negative and neutral examples. Positive examples are the true exemplars. Neutral ones are the examples, which share characters at certain positions. For example, for a query word of “search”, the word “series” is considered as neutral, since both of them share “se”. Since we are using fixed length representations built out of profile features, this minimizes contradictions. Without introduction of a neutral class, learning becomes difficult. All examples which are neither positive nor neutral are negative examples.

3.4.1 Retrieval with learnt distance functions

Our retrieval approach learns the query specific distance function for optimizing the rank. The iterative method described in the previous section can be done in closed form for a completely annotated data set. This also removes the necessity of having a user to refine the results. Learning distance functions from labelled examples have received considerable attention in the recent past [86, 10].

Table 3.3 presents a quantitative comparison of performance enhancement with learning. We can see that learning with Equations 3.1 or 3.2 are almost similar. Figure 3.7 shows qualitative examples of the retrieval with and without learning. It can be seen that visually similar looking images are retrieved with no learning, and with learning a better set is obtained.

Datasets	No Learning	QSC with Eq 3.1	QSC with Eq 3.2
CD	0.737	0.946	0.944
RD	0.817	0.930	0.939

Table 3.3 Results(mAP) on two Dataset with 300 queries in each.

Some of the state of art approaches learn a Mahalanobis distance metric for the k- nearest neighbor classification by semidefinite programming [86]. The metric is learnt such that top k- neighbors always belong to the the same class while examples from the different classes are separated by a larger margin. Some of others learn a distance function for each training image as a combination of elementary distances between patch-based visual features. Then apply these to classification of novel images [26]. There is another approach where the weights are the ratios of standard deviations of the complete dataset to the the images selected by user, but in our case we do not use neutral examples and also their method is restricted towards new unseen queries. Learning has been used in document retrieval, one of such method is [44]. In their work they also aim to search for keywords given by the user in a large collection of digitized printed historical document and retrieve a better set with the help of user feedback but again their method doesn't use any distance function learning.

3.5 Extrapolation of Query Specific Classification

In the previous section we presented methods which allow us to learn better distance functions for a given query. However, this method is limited to a finite set of query images. In a generic setting of document image retrieval, user would like to give an arbitrary query and retrieve the most relevant document images based on word similarity computation. We support textual queries for this purpose.

We enable users to provide user queries (rather than already existing word images) by rendering queries and generating word images. Even then the retrieved results need not be accurate because the distance functions are not designed for these queries. We would like to extrapolate the learnt distance functions into a novel setting.

It is known that when the database is diverse, retrieval performance deteriorates. This is due to the fact that database contains multiple (possibly unknown) fonts and styles. To address this issue we extrapolate the learnt distance function to novel content and style space without any user involvement. The weight vector learnt for similar queries (where substrings are same), will be similar in dimensions corresponding to the common substrings. We use this fact and synthesize new weight vectors which share substrings with the query word in the training database. To enhance the performance in a heterogeneous style collection we retrieve words based on multiple distance functions and generate a final list by merging the multiple set.

In real life database we have variations in style and content. However, the number of possible variations and styles (say fonts, sizes) is far less compared to the content variations. It has been shown in the past that given a collection, style and content can be separated [83] with the help of expectation maximization (EM) algorithm. However, a purely unsupervised distance function learning in such a framework will be computationally intensive. That is why we resort to the method of learning appropriate distance function from the training dataset and extend to the test dataset.

Here, distance functions are learnt for each word in the RD dataset using the method described in Section 3.4. This is done in closed form using the groundtruth word images in the training set. Since the feature descriptors also depend on the style variations of the text we do style specific learning. Hence for each word image w_i and style s_k we have a word weight vector $W_w(s_k, w_i)$. During retrieval, from the unseen textual word query, word images with the same text as query are synthesized in different styles. Distance functions for these synthetic word images are computed from already learnt weight vectors by extrapolation. We achieve this by disintegrating the learnt weight vectors for words into learnt weights vectors for subwords during training. And then by using these subword weight vectors to compute weight vectors for unseen words.

Since the feature vectors are extracted using a sliding window, the descriptor depends on the width of the image. Therefore, each subword's contribution to the word descriptor (and eventually word weight vector) is directly proportional to the width of that subword in the word. A subword can be a pair or a sequence of characters or just a character, here we refer to character as subword. For any style we know the approximate relative width for each subword in the word. Now, we divide the word weight vector $W_w(s_k, w_i)$ according to relative widths of the subwords to get weight vectors for each of them. These subword weight vectors are then scaled to have a constant dimension l_d. For style s_k, final weight vector for a subword l_j, $W_l(s_k, l_j)$, is computed as:

$$W_l(s_k, l_j) = \sum_{i \in W_{l_j}} \frac{W_{wl}(s_k, l_j, w_i)}{|W_{l_j}|} \tag{3.3}$$

where W_{l_j} is set of all words that contain l_j and $W_{wl}(s_k, l_j, w_i)$ is the weight vector for subword l_j when it occurs in word w_i and style s_k. We call this *content specific disintegration* of word weight vector or *construction* of subword weight vector. This is demonstrated in Figure 3.8(a). We take an example of a word "above" from RD dataset. It is shown how with the help of query specific learning method we learn a weight vector for this word image and then disintegrate it to get subword weight vectors. Please note that the subword weight vectors are subdivisions taken from word weight vector based on the relative subword width, which is approximate and not exact. Inspite of this, our method achieves good precision that shows its robustness. The projected subword weight vectors, $W_{wl}(s_k, l_j, w_i)$, are shown at the end of the pipeline. Such weight vectors are computed for each subword-style combination across all the word images in the dataset. And the final subword weight vector computed using Equation 3.3.

Now as the output of the above training phase we have style and content dependent weight vectors for each style-subword combination. Essentially, we learn the matrix,

$$M(k, j) = W_l(s_k, l_j), \quad k = 1, 2...|S| \quad and \quad j = 1, 2...|C|$$

where S is set of all styles present in the database and C is set of all characters or subwords.

This matrix, M, of weight vectors is used to extrapolate the learning from training data to get weight vector for any new/unseen word. Specifically, features are extracted from the synthetic test word images for each style. Then learnt weight vectors for each of the subwords in the test word image are mapped from l_d to a new dimension. These new lengths are directly proportional to the relative lengths of the corresponding subwords and chosen such that their sum is equal to d. These are appended to get the d dimensional weight vector of the query word image for that particular style. This process of extrapolation for retrieval is shown in Figure 3.8(b). As an example we take a textual query "close" from the UD dataset.

We perform two experiments to evaluate our query specific learning with extrapolation method. In the first one, we test our method by retrieving from a dataset which is same as the training dataset. For this purpose, we train using CD, RD datasets and retrieve from the respective dataset. During evaluation, we selected 300 queries from each of the datasets and retrieved 30 images per query with the smallest distance. The mean of average precisions are reported in first and second row of the Table 3.4.

Dataset	Measure	DTW	Euclidean	QSC with extrapolation
CD	mAP	0.853	0.764	0.902
RD	mAP	0.778	0.817	0.923
UD	mP	0.890	0.915	0.955

Table 3.4 Comparative results of extrapolation on CD, RD and UD datasets.

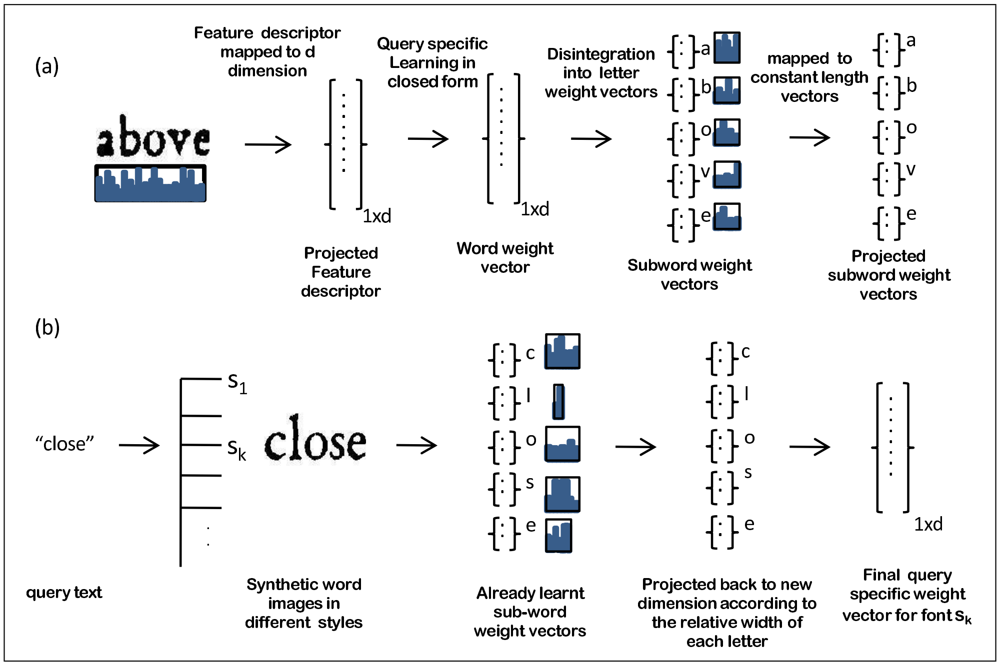

Figure 3.8 (a) The pipeline shows how a weight vector is learnt for each subword during training in RD dataset and (b) The pipeline shows how a weight vector is generated by extrapolation for an unseen query which is later used for retrieval.

In the second experiment, we test our method by retrieving from a dataset which is different from the training dataset. For this, we train using RD dataset and retrieve from UD dataset. Since UD dataset is unannotated, we do not know the total number of true positives and therefore cannot compute average precision. Therefore, we compute mean precision as a performance measure which is shown in last row of the Table 3.4. We use 25 selected queries for this experiment and retrieved 30 images per query. We also show its qualitative results on three queries in Figure 3.9. Variation in the top 10 retrieved results shows that our method is robust towards variation in fonts and styles. In both the experiments our method outperforms the other two and also shows good generalization.

The above method of extrapolation involves projecting a feature descriptor for word images to lower or higher dimension feature space (d dimension). Projecting to a very different feature dimension may lead to loss of important information. So it is important to set the value of d appropriately. We conduct an experiment to observe the effect of d on the retrieval performance. Figure 3.10 shows how mAP varies with d. From the plot, it can be observed that the performance is stable for a good range: $d = 110$ to $d = 270$. The range of d depends on the database which means that it can be different for different languages.

The extrapolation method presented in this section enables generalization of the learning done from training data. It makes it possible to generate weight vector for any unseen word.

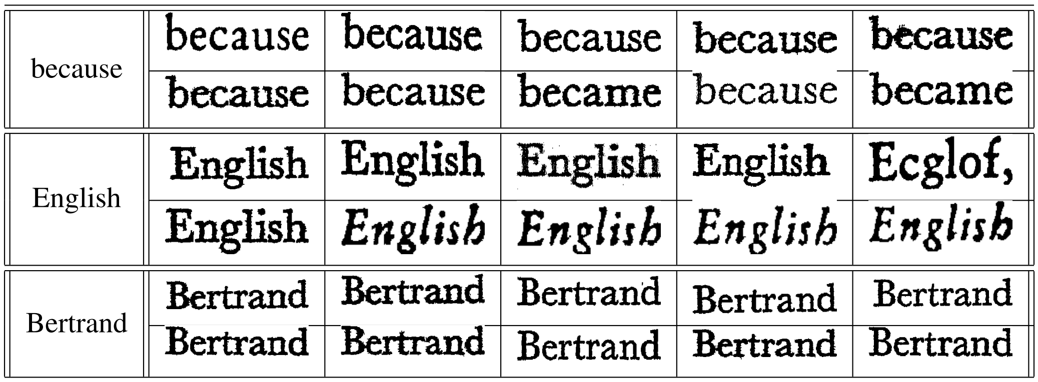

Figure 3.9 Examples of word images retrieved from unannotated database of more than 5M words. First column shows the query word. Some of the retrieved results are shown. For the top-20 retrieval, mean of precision is 0.936

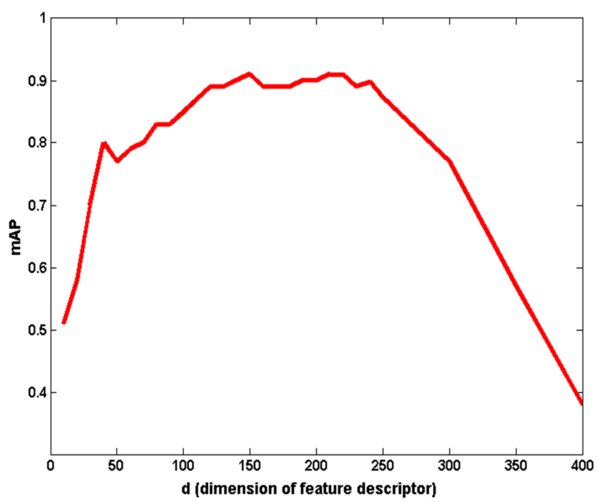

Figure 3.10 Graph shows how mAP changes when dimension of feature vectors increases.

3.6 Summary

In this chapter, a word image matching scheme based on fixed length representation is presented. The method uses a close form of relevance feedback technique for learning character weights. These learnt weights are being used for constructing the weight vector for a query word. The method showed better performance over reference systems based on DTW and Euclidean distance measure.

Chapter 4

BLSTM Based Word Retrieval for Hindi Documents

Retrieval from Hindi document image collections is a challenging task. This is partly due to the complexity of the script, which has more than 800 unique ligatures. In addition, segmentation and recognition of individual characters often becomes difficult due to the writing style as well as degradations in the print. For these reasons, robust OCRs are non existent for Hindi. Therefore, Hindi document repositories are not amenable to indexing and retrieval. In this chapter, we propose a scheme for retrieving relevant Hindi documents in response to a query word. This approach uses BLSTM neural networks. Designed to take contextual information into account, these networks can handle word images that can not be robustly segmented into individual characters. By zoning the Hindi words, we simplify the problem and obtain high retrieval rates. Our simplification suits the retrieval problem, while it does not apply to recognition. Our scalable retrieval scheme avoids explicit recognition of characters. An experimental evaluation on a dataset of word images gathered from two complete books demonstrates good accuracy even in the presence of printing variations and degradations. The performance is compared with baseline methods.

4.1 Introduction

Word spotting and document image retrieval have received significant attention in recent years. The success of word spotting can be attributed to the holistic matching paradigm employed for comparing a query word with the images in the database. Both the query as well as the database images are converted into a sequence of feature vectors and comparison is often carried out with the help of Dynamic Time Warping (DTW) or other matching scheme. This feature representation for the whole image avoids an explicit recognition step. Such word matching schemes have applications in accessing historic handwritten manuscripts [72], searching for relevant documents in a digital library of printed documents [9], and holistic recognition [48].

Word spotting becomes applicable to printed documents in a variety of situations. For the task of document retrieval, word spotting based methods are favored over direct character recognition. OCR systems do not perform well on degraded documents, which leads to an insufficient performance for a robust retrieval systems. This becomes even more apparent when working on Indian scripts where robust OCR systems are not available to the knowledge of the authors.

The Devanagari script for writing Hindi requires as many as 800 classes for recognition [63] and therefore a large amount of training data is required to build a robust recognition system. Many of these classes are formed by adding vowel modifiers consisting of a stroke at the top or at the bottom of a consonant. Although words can not be correctly recognized without the modifiers, we show that they can be ignored for keyword spotting based document retrieval for Hindi. Therefore, we dramatically reducing the amount of training data needed. To disambiguate words which differ only in the upper and lower vowel modifiers, a post processing step is applied.

We use BLSTM neural networks for word spotting in this chapter. This category of neural networks was previously used for speech recognition [77], handwriting recognition [30], as well as keyword spotting for English handwritten text [25]. We extend these methods for wordspotting in printed Hindi documents. BLSTM neural networks are recurrent networks with hidden layers consisting of long short-term memory blocks. BLSTM neural network transform a sequence of feature vectors into a sequence of character class probabilities, and thereby building an intermediate representation which is used for word spotting. Another common technique for sequence recognition are Hidden Markov Models (HMMs). However, on a similar keyword spotting task they are outperformed by BLSTM based keyword spotting systems [25].

In this chapter, we demonstrate the successful use of a BLSTM neural network for printed Hindi documents. Our focus is on retrieval from a collection of documents written in Devanagari, a script for which robust and reliable OCR system do not exist. Our method takes an intermediate path between the complete recognition, which HMMs and BLSTM neural networks are capable of, and matching of feature vectors (popular in traditional word spotting literature) which does not scale to large data sets. We demonstrate our method on a couple of printed books in Hindi.

The chapter is organized as follows. We first present the basic properties of the Devanagari script, (used to write the Hindi language) in Section 4.2. We also discuss previous attempts at recognition of this language. We then describe our retrieval scheme based on BLSTM neural network in Section 4.3. The performance of the proposed scheme is reported in Section 4.4. In Section 4.5, we use BLSTM neural network as recognizer and show the word and character level accuracy on Hindi documents.

4.2 Hindi and Devanagari

India is the second most populous country in the world, with a population of over one billion, and is the seventh largest country by geographical area. It is a home to some of the most ancient civilizations, and a center of important historic trade routes. Devanagari is the most popular script in India. Hindi, the national language of India is written in Devanagari and has been adopted along with English as the oftencial language of the central government. It is the third most widely spoken language in the world [65] and has more than 337 million speakers. Hindi is written from left to right. It does not have

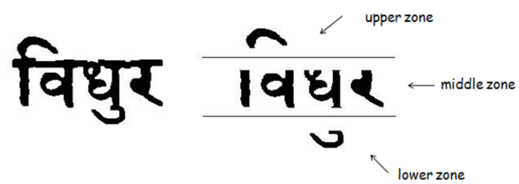

Figure 4.1 An example word in Devanagari and its upper, middle, and lower zone

distinct letter cases. Words are often connected with the help of a head line — *sirorekha*. The script contains basically 11 vowels and 33 consonants. Vowels can be written independently, or combined with a consonant to form conjuncts (above, below, before or after). Vowels written in this way are called modifiers or *matra*. Sometimes multiple consonants and vowels are combined to form a new shape which are termed as compound characters. The number of such compound characters can, in principle, be a few thousand. However, in practice, there are around 800 unique ligatures in this language [63]. The exact number depends on the font and the rendering scheme used.

There may not be a sufficient number of examples of each class present in the training set. To avoid incorrect retrieval we adopt the following strategy. Since, the words are characterized by a connecting *sirorekha*, one of the popular initial steps in recognizing Hindi is to remove this line. This separates the vowel modifiers which span the top-zone from the middle (main) zone. This is pictorially explained in Figure 4.1. Similarly the lower zone is also removed. This reduces the number of character classes to be recognized. Most of the previous work in the recognition of Hindi documents follow this approach or a variant of this for simplifying the class list [43, 14].

One of the earliest work in recognizing Hindi document is from Chaudhuri and Pal [14]. There have also been attempts in recognizing Hindi documents in recent years [11, 43]. In general, all these methods were based on an explicit segmentation of the words into characters. A major alternative to this, is the segmentation free method followed by Natarajan *et al.* [63]. They use an HMM based classification scheme for recognition of words and lines directly.

In general the number of character classes can be anywhere from 300 to 800 [63, 14]. The level of segmentation controls the number of classes to be handled. In this work, our objective is not a complete recognition of Hindi documents. We are interested in spotting words, or retrieving relevant Hindi documents corresponding to a query. We take an intermediate path between complete recognition and feature based word spotting in this work.

4.3 Neural Network based Word Spotting

In this chapter, we perform word spotting based on a recently developed recurrent neural network, termed bidirectional long short-term memory (BLSTM) neural network [30]. We avoid a complete recognition of the Hindi documents here, due to the computational and script related challenges associated with the problem. BLSTM is a bidirectional recurrent neural network (BRNN) that has hidden layers, which are made up of the so-called long short-term memory (LSTM) cells. These networks are capable of processing a finite sequence, and predict a label sequence based on both the past and the future context of the element. BLSTM has many advantages for document recognition and retrieval [25]. It avoids explicit segmentation of words into characters, which are difficult in many situations due to the degradations present in the data set.

A major obstacle in using recurrent networks has been that error gradients vanish exponentially with the size of the time lag between important events. This is called the vanishing gradient problem. Using LSTM hidden layers, BRNN neural networks address this problem by controlling the information flow into and out of each memory blocks by nodes [30]. A way to circumvent this problem is the introduction of so-called long short- term memory blocks. In Figure 4.2 such a LSTM node is displayed. At the core of the node, a simple cell, which is connected to itself with a recurrence multiplication factor of 1.0 stores the information. New information via the Net Input enters only if the Input Gate opens and leaves the cell into the network when the Output Gate is open. The activation of the Forget Gate resets the cells value to 0. The gates and the Net Input are conventional nodes using an arctan activation function. This architecture admits changes to the cells memory only when one of the gates is open and is therefore able to carry information across arbitrarily long sequence positions. Thus, at any point in the sequence, the usage of contextual information is not restricted to the direct neighborhood.

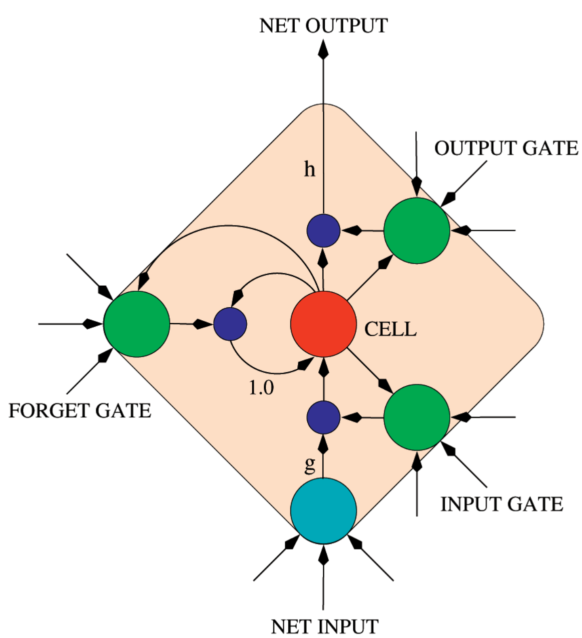

Figure 4.2 The LSTM node

We represent the word image as a sequence of feature vectors of dimension d. In our implementation, we use $d = 5$ features. BLSTM neural networks contain one node for each of the features in the input layer. Thus the network has d input layer nodes. Each node in the input layer is connected with two separate hidden layers, one of which processes the input sequence of features forward, while the other processes it backward. Both hidden layers are connected to the same output layer. The output layer contains one node for each character class plus a special node ε, to indicate "no character". Thus, There are $K+1$ nodes in the output layer, where K is the number of classes, in our case $K = 81$. More details describing the functioning of BLSTM neural network can be seen in [29].

For each element position in the input feature sequence, the hidden layers provide access to the past as well as future context to the output layer. The output layer sums up the values which comes from both forward and backward hidden layers. These output activations of the nodes in the output layer are normalized (such that they sum up to one) and then treated as a probability vector of the characters present at the position. All the probability vectors together form a matrix of character probabilities, as shown in Figure 4.3. The dark cells show high probabilities and the probability is one when it is completely black. Similarly, light cells means low probabilities and the probability is zero when the cell is completely white. If there are W samples in the input feature sequence, the matrix has a dimension of $(K+1) \times W$. Each path through this matrix which corresponds to a possible character sequence. The overall probability of the sequence is directly given by the product of all elements along its path. Note, however, that such a naive decoding will not result in a semantically meaningful character decoding.

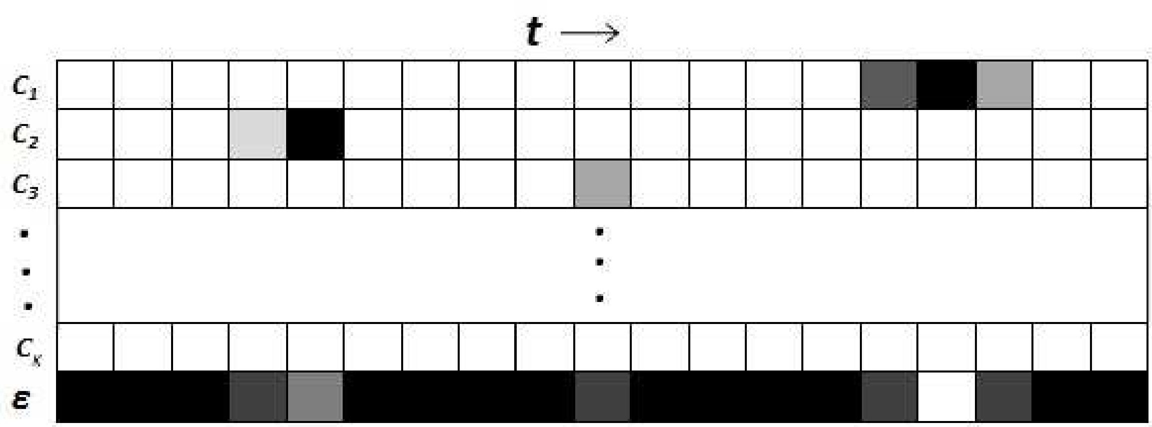

Figure 4.3 The sequence of output activations of the neural network.

The most likely sequence of characters is given by choosing the character with the highest probability for each step in the sequence. Afterwards, this character sequence needs to be shortened. A single character in the word image can lead to multiple subsequent activations of the same node. Also, activations of the output node corresponding to ε indicate that no decision about a character can be made at that position. To transform a path to a character sequence, all subsequent occurrences of the same character are merged and then the ε symbol is deleted. Hence, the two paths $c_1, \varepsilon, \varepsilon, c_2, c_2, \varepsilon$ and $\varepsilon, c_1, c_1, c_2, c_2, \varepsilon$ indicate the same character sequence c_1c_2. For more details on BLSTM neural networks please refer to [30].

We work with word images instead characters and lines. This is primarily motivated by the fact that we are interested in retrieving documents based on queries which are primarily words. First, document images are segmented into words. Since the *sirorekha* connects the whole word and increases the number of possible shapes to be recognized, we segment the word by removing *sirorekha*. This is done with the help of a horizontal projection. We remove the top and bottom zone and work with the middle zone. This can result in lowering of the precision at the same recall. However, this reduction is not very significant. In addition, with a simple re-ranking, we overcome most of the loss in precision incurred due to this. We manipulate the ground-truth label sequence of the images to obtain the corresponding ground-truth after zoning it in the middle. After removing upper and lower zone, each image is represented as a sequence of feature vectors. This does not require any further segmentation of the word images. This makes the process robust to the typical cuts and merges seen in the document images.

Word images are represented as a sequence of feature vectors. For all our experiments, we use features extracted from vertical stripes of uniform length (sliding window). We extract five different features, (a) the lower profile, (b) the upper profile, (c) the ink-background transitions, (d) the number of black pixels, and (e) the span of the foreground pixels. The upper and lower profiles measure the distance of the top and bottom foreground pixel from the respective baselines. Ink-background transitions measures the number of transitions from Ink to background and reverse. the number of black pixels provides the information about the density of ink in the vertical stripe.

These input sequences of feature vectors are fed to the network for training together with their corresponding label sequences. Once the system is trained, it uses the neural network output probabilities to return the most likely label sequence for given test sequence. Then, all database images are stored as the corresponding output label sequence returned by the neural network. For word retrieval, these output labels are compared with the query image's label sequence using edit distance and ranked based on this score.

4.4 Results

In this section, we present experimental results to validate the utility of the proposed method. We also quantitatively compare the performance against baseline methods. To evaluate the performance, we have conducted two experiments. For the first experiment, we consider a complete book (we call this book as *book1*) for training and testing. For the second experiment, we train on *book1* and test on a different book (*book2*). Both these books are annotated at the word level [39]. Details of these two books are given in Table 4.1.

In the first experiment, the document images were scanned and segmented into word images. We used 60% of the images as the training set and 20% as the validation set. The remaining 20% was used as a test set. We trained the network using the training set and used the validation set to decide the stopping criteria during the iterative training process. Each word image in test set is given to the trained network as an input. The network outputs a most probable sequence of labels for each image

Book	#Pages	#Lines	#Words
book1	98	2463	27764
book2	108	2590	28265
book3	93	2289	22244
book4	85	2599	24265
book5	118	3023	33989
book6	123	3232	34002

Table 4.1 Books used for the experiments.

with the help of output probabilities. This label sequence is further used to compute the edit distance from the query words for word retrieval. To show the word retrieval performance, we selected two sets of frequently occurring keywords. The first set contains words which are already present in the training set (in-vocabulary) and the other set contains words which are not present in the training set (out-of-vocabulary). Both these sets have 100 queries each. These key words are selected such that they (i) have multiple occurrences in the data set (ii) have mostly functional words and (iii) have very few stop words. The results of in-vocabulary and out-of-vocabulary queries are shown separately in Table 4.2. Here P represents the precision at a recall rate of 50%. Mean Average Precision (MAP) is the mean of the area under the Precision-Recall curve for all the queries.

Queries	P	MAP
in-vocabulary	92.90	84.18
out-of-vocabulary	92.17	82.91

Table 4.2 Results of BLSTM based method on in-vocabulary and out-of-vocabulary queries (100 each).

Method	P	MAP
DTW	84.64	77.39
Euclidean	78.23	71.82
BLSTM based	91.73	84.77
BLSTM with reranking	93.26	89.02

Table 4.3 A comparison with the reference systems.

In the second experiment, we trained the neural network on *book1* and tested on *book2*. We selected 100 most frequently occurring function keywords from the test set as queries. For retrieval, we compute their edit distances from the rest of the test images using their label sequence representations. We then ranked the retrieved results based on their distances from the query keyword and computed the MAP. We also compute the precision at a recall rate of 50%. We obtained MAP of 84.77% in this experiment. We observed that most confusions come from dissimilar words with similar middle zones. To solve this issue, we do a re-ranking with the help of the upper zone. We re-rank the retrieved set based on the

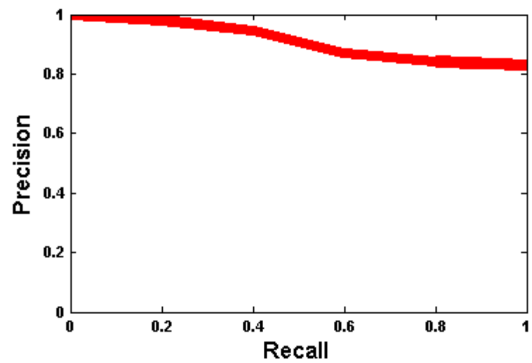

Figure 4.4 Precision-recall curve evaluated for 100 queries.

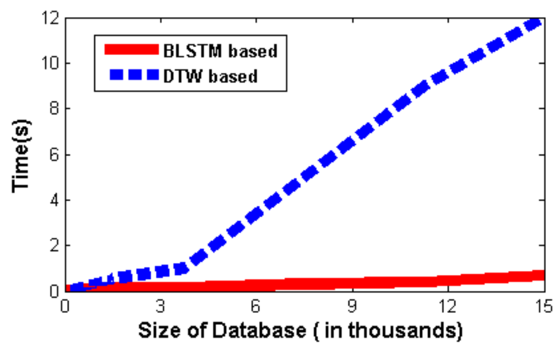

Figure 4.5 This graphs compares the average time takes by two different methods, BLSTM and DTW distance, per query on increasing database size.

number of connected component present in the upper zone to avoid dissimilar images with the same middle zone. This scheme does not depend on the size or font of characters in the text word. The use of simple features, which are easy to compute, make our system fast and robust. Results of our method are shown in the last two rows of Table 4.3. The precision-recall curve corresponding to the BLSTM with re-ranking is shown in Figure 4.4.

Traditional word spotting/retrieval methods are also applicable for Hindi documents. We compare our results with word spotting methods with DTW and Euclidean distance. In both cases, the the feature vectors are the same. For Euclidean, all words are normalized to the same width (145 pixels in our case). We show the comparison of the neural network based method with DTW distance and Euclidean distance based approach in the Table 4.3. It may be seen that the proposed methods are superior in retrieving relevant words.

Finally, we compare the computational complexity of the proposed retrieval scheme with the DTW based baseline. In both the cases, a dynamic programming based retrieval is carried out. However, the feature representation is different. The plot in Figure 4.5 compares the retrieval time for both neural

Figure 4.6 First column shows the query words. Their retrieved images are shown according to the rank in row major.

network method and DTW based retrieval. It is clearly seen that the neural network based method is faster than the DTW based matching and search procedure.

We have also shown the qualitative results on five different queries in Figure 4.6. Variation in the retrieved results shows that the network based method is robust towards degradations and print variations. Training on *book1* and testing on *book2* shows some level of generalization across print variations. However, more work is required for demonstrating font and style independence across a larger spectrum.

Hence to show the proposed scheme's performance on larger dataset, we have choosen a set of 6 books of different font styles and sizes. We chose each book as training and tested on the remaining books. We did this for every book. The results of this experiments shown in Table 4.4.

Method	Measure	book1	book2	book3	book4	book5	book6
NN	P	83.23	85.44	82.99	79.37	80.43	81.97
NN	mAP	70.91	73.18	71.92	68.32	67.32	70.83
NN with PP	P	83.01	85.49	85.11	81.62	84.06	84.02
NN with PP	mAP	75.43	79.86	73.59	73.64	72.80	74.91

Table 4.4 Results of BLSTM based methods (100 queries each).

We also conducted the experiments by refining the classlist. We have added some classes which are less frequent and modeled by combing two or more characters together. The Table 4.5 contains this experiment's result.

Till now we have limited our experiments with only 5 features and shown the results. We have taken some more features into consideration and evaluated the performance with them. We added 3 more

Method	P	mAP
NN	91.73	84.23
NN with PP	93.16	89.22

Table 4.5 Results of BLSTM based methods after refinement in the classlist.

features 1st moment, 2nd moment and 3rd moment. The Results are shown in Table 4.6 after adding these features.

Method	added F1	added F2	added F2	added all	old feat
NN	83.90	79.56	82.66	82.92	84.77
NN with PP	87.69	88.96	87.54	88.69	89.02

Table 4.6 Results of BLSTM based methods with different features.

4.5 Exhaustive recognition using BLSTM

In this section, we attempt to use BLSTM neural network as a recognizer. Here, we do not remove upper and lower zone of the word images, rather we feed word images directly to the neural network as inputs. The training data has ground truth associated with it and used for training the network. Feature extraction method is same as what we have been using so far in this chapter.

To test BLSTM as recognizer, we train the network with 3 books, containing 66509 word images in total. We use 2 books, containing 23345 word images as validation set (used to stop the back-propagation iterations in the training process). We then use the trained network on a new book (not used in either training or validation data) for testing. Network took 216 epochs to reach convergence point. On testing data, character accuracy rate of 90.57% and word accuracy of 76% are observed.

Exhaustive experiments on recognition using BLSTM neural network is not in the scope of this thesis. However, we believe that BLSTM neural network based recognition is a promising area and can be used for recognizing challenging Indian languages in upcoming years.

4.6 Summary

In this chapter, a word retrieval scheme for Hindi documents is presented. It bases on two principles. First, instead of an OCR system, a sequential approach to word recognition in followed. Words are stored as possible character sequences computed by BLSTM neural networks and the edit distance of this sequence to the query word is used. Secondly, in order to circumvent the problems that arise due to the large number of possible character classes encountered in Devanagari script, vowel modifications are removed. This reduces the number of classes drastically. Disambiguations that may arise are resolved in a post-processing step. This way, the proposed method takes an intermediate path between the complete recognition and feature based word spotting task.

On a set of experiments using printed Devanagari script, the superiority of the proposed method over a range of reference systems can be shown. The success originates from a well-balanced system. On the one hand, we adopt a sophisticated recognition technique for the retrieval task. On the other hand, by avoiding a complete recognition, we bypass the challenges posed by the script.

We then used BLSTM neural network as a recognizer and shown an acceptable word/character level recognition accuracy. We use complete list of class list here and did exhaustive recognition.

Chapter 5

Conclusions

As enormously large quantity of documents are getting archived by many applications including the digital libraries, access to the content of such collections becomes a challenging task. We attempt to design schemes for document image recognition as well as document image retrieval so as to facilitate access to digitized document collections. In India there are a number of languages with their own indigenous scripts. Accordingly, there is a bulk of printed materials available in libraries, information centers, museums and other institutions. But, the document analysis and understanding area has been given less emphasis for these languages in the recent past.

In this thesis, we explore different approaches for addressing the document analysis and understanding problems with emphasis on English and Indian languages. After extensive script analysis, we have designed a retrieval system for english and hindi printed document images. We have conducted experiments to evaluate its performance, in which we have obtained significantly good results on reasonably diverse quality documents.

We have argued that a matching scheme based on fixed length representation can be learnt to enhance the word image matching module. We demonstrated that a query specific classification algorithm can be designed for retrieval of word images. This can be done in a computationally efficient manner. Experimental results show the effectiveness of our design for the retrieval from English document image collections. This strategy is promising for the retrieval of large digitized document images in many applications, including digital libraries.

Furthermore, a novel keyword spotting system for Hindi documents was developed as part of this thesis. In an extensive experimental setup, its superior performance over existing state-of-the-art systems could be shown on printed data. Finally, we used this BLSTM recurrent neural network for recognizing Hindi word images without character level segmentation.

Plenty of promising and important research directions can be followed from here. First, more recognition-free search systems can be involved. Secondly, a more detailed research can be done using BLSTM NN in order to build a recognition system. Future direction also includes the development of a scalable system which can retrieve word images by implicitly using language models.

The matching schemes (designed in this work) needs more engineering effort to make them applicable for real-life situations. To this end, we recommend an extension work to develop an online system for searching hundreds and thousands of books over the Web.

Our retrieval and recognition work is more concentrated on scanned printed documents. For better visualization of the large image collections at hand, we recommend to further the research work for the recognition and retrieval from complex documents (such as camera-based, handwritten, etc).

Publications

- Raman Jain, Volkmar Frinken, C. V. Jawahar, R. Manmatha
  BLSTM Neural Network based Word Retrieval for Hindi Documents
  In Proceedings of the IEEE International Conference on Document Analysis and Recognition (ICDAR), Beijing, China, 2011.

- Raman Jain, C. V. Jawahar
  Towards More Effective Distance Functions for Word Image Matching
  In Proceedings of the IAPR Document Analysis System (DAS), Boston, U.S. 2010.

- Praveen Dasigi, Raman Jain, C. V. Jawahar
  Document Image Segmentation as a Spectral Partitioning Problem
  In Proceedings of the IEEE International Conference on Computer Vision, Graphics and Image Processing (ICVGIP), Bhubaneswar, India, 2008.

Bibliography

[1] Digital library of india. http://dli.iiit.ac.in/.

[2] The universal library. http://www.uliborg.

[3] Wikipedia. http://en.wikipedia.org/wiki.

[4] C. C. Aggarwal. Towards systematic design of distance functions for data mining applications. In *KDD*, pages 9–18, 2003.

[5] S. Aksoy, R. M. Haralick, F. A. Cheikh, and M. Gabbouj. A weighted distance approach to relevance feedback. In *ICPR*, pages 4812–4815, 2000.

[6] A. Andreev and N. Kirov. Word image matching based on hausdorff distances. In *ICDAR*, pages 396–400, 2009.

[7] E. Ataer and P. Duygulu. Retrieval of ottoman documents. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 152–162, 2006.

[8] D. Bainbridge, J. Thompson, and I. Witten. Assembling and enriching digital library collections. In *Digital Libraries, 2003. Proceedings. 2003 Joint Conference on*, pages 323–334, may 2003.

[9] A. Balasubramanian, M. Meshesha, and C. V. Jawahar. Retrieval from document image collections. In *Document Analysis Systems*, pages 1–12, 2006.

[10] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[11] A. Bhardwaj, S. Kompalli, S. Setlur, and V. Govindaraju. An ocr based approach for word spotting in devanagari documents. In *DRR*, page 68150, 2008.

[12] C. Böhm, S. Berchtold, and D. A. Keim. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys.*, 33(3):322–373, 2001.

[13] J. Chan, C. Ziftci, and D. Forsyth. Searching off-line arabic documents. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1455–1462, Washington, DC, USA, 2006.

[14] B. B. Chaudhuri and U. Pal. An ocr system to read two indian language scripts: Bangla and devnagari (hindi). In *ICDAR*, pages 1011–1015, 1997.

[15] B. B. Chaudhuri, U. Pal, and M. Mitra. M.mitra, automatic recognition of printed oriya script. *Sadhana*, 27:23–34, 2002.

[16] S. Chaudhury, G. Sethi, A. Vyas, and G. Harit. Devising interactive access techniques for indian language document images. In *ICDAR*, pages 885–889, 2003.

[17] S. Chaudhury, G. Sethi, A. Vyas, G. Harit, and V. G. Harit. Devising interactive access techniques for indian language document images. In *in Proc. International Conference on Document Analysis and Recognition*, pages 885–889, 2003.

[18] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *VISSAPP (2)*, pages 273–280, 2009.

[19] T. Deselaers, R. Paredes, E. Vidal, and H. Ney. Learning weighted distances for relevance feedback in image retrieval. In *ICPR*, pages 1–4, 2008.

[20] D. S. Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.

[21] N. Doulgeri and E. Kavallieratou. Retrieval of historical documents by word spotting. In *Document Recognition and Retrieval*, pages 1–10, 2009.

[22] O. Due Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition-a survey. *Pattern Recognition*, 29:641–662, 1996.

[23] J. Edwards, Y. W. Teh, D. A. Forsyth, R. Bock, M. Maire, and G. Vesom. Making latin manuscripts searchable using ghmms. In *NIPS*, 2004.

[24] A. Franco, A. Lumini, and D. Maio. A new approach for relevance feedback through positive and negative samples. In *ICPR (4)*, pages 905–908, 2004.

[25] V. Frinken, A. Fischer, H. Bunke, and R. Manmatha. Adapting blstm neural network based keyword spotting trained on modern data to historical documents. In *ICFHR*, pages 352–357, 2010.

[26] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *NIPS*, pages 417–424, 2006.

[27] B. Gatos and I. Pratikakis. Segmentation-free word spotting in historical printed documents. *Document Analysis and Recognition, International Conference on*, 0:271–275, 2009.

[28] A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *NIPS*, 2005.

[29] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[30] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(5):855–868, 2009.

[31] D. Greenstein and S. E. Thorin. The digital library: A biography, 2002.

[32] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh. *Feature Extraction, Foundations and Applications*. Springer, Inc., 2006.

[33] E. Hassan, S. Chaudhury, and M. Gopal. Shape descriptor based document image indexing and symbol recognition. In *ICDAR*, pages 206–210, 2009.

[34] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification and regression. In *NIPS*, pages 409–415, 1995.

[35] T. Hong, T., and J. J. Hull. Algorithms for postprocessing ocr results with visual inter-word constraints. In *ICIP*, 1995.

[36] R. D. S. II and K. Fukunaga. The optimal distance measure for nearest neighbor classification. *IEEE Transactions on Information Theory*, 27(5):622–626, 1981.

[37] Y. Ishitani. Model-based information extraction method tolerant of ocr errors for document images. In *ICDAR*, pages 908–915, 2001.

[38] A. K. Jain and A. M. Namboodiri. Indexing and retrieval of on-line handwritten documents. In *ICDAR*, pages 655–, 2003.

[39] C. V. Jawahar and A. Kumar. Content-level annotation of large collection of printed document images. In *ICDAR*, pages 799–803, 2007.

[40] C. V. Jawahar, M. N. S. S. K. P. Kumar, and S. S. R. Kiran. A bilingual ocr for hindi-telugu documents and its applications. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 1*, ICDAR '03, pages 408–412, 2003.

[41] C. V. Jawahar, M. Meshesha, and A. Balasubramanian. Searching in document images. In *ICVGIP*, pages 622–627, 2004.

[42] A. Kolcz, J. Alspector, M. Augusteijn, R. Carlson, and G. V. Popescu. Visual keyword based word-spotting in handwritten documents. In *SPIE International Symposium on Document Recognition*, pages 26–30, 1998.

[43] S. Kompalli, S. Setlur, and V. Govindaraju. Devanagari ocr using a recognition driven segmentation framework and stochastic language models. *IJDAR*, 12(2):123–138, 2009.

[44] T. Konidaris, B. Gatos, K. Ntzios, I. Pratikakis, S. Theodoridis, and S. J. Perantonis. Keyword-guided word spotting in historical printed documents using synthetic data and user feedback. *IJDAR*, 2007.

[45] T. Konidaris, B. Gatos, S. J. Perantonis, and A. Kesidis. Keyword matching in historical machine-printed documents using synthetic data, word portions and dynamic time warping. *DAS, IAPR International Workshop*, 2008.

[46] A. Kumar, C. V. Jawahar, and R. Manmatha. Efficient search in document image collections. In *ACCV (1)*, pages 586–595, 2007.

[47] P. Langley. Selection of relevant features in machine learning. In *In Proceedings of the AAAI Fall symposium on relevance*, pages 140–144. AAAI Press, 1994.

[48] V. Lavrenko, T. M. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *DIAL*, pages 278–287, 2004.

[49] M. Lesk. Practical digital libraries: Books, bytes, and bucks. 1997.

[50] J. Li, Z.-G. Fan, Y. Wu, and N. Le. Document image retrieval with local feature sequences. In *ICDAR*, pages 346–350, 2009.

[51] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[52] D. G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7:72–85, 1995.

[53] Y. Lu, C. L. Tan, W. Huang, and L. Fan. An approach to word image matching based on weighted hausdorff distance. *ICDAR*, 2001.

[54] Y. Lu, C. L. Tan, and S. Member. Information retrieval in document image databases. *IEEE Transactions on Knowledge and Data Engineering*, 2004.

[55] R. Manmatha, C. Han, and E. M. Riseman. Word spotting: A new approach to indexing handwriting. In *CVPR*, pages 631–637, 1996.

[56] S. Marinai, E. Marino, F. Cesarini, and G. Soda. A general system for the retrieval of document images from digital libraries. In *DIAL*, pages 150–173, 2004.

[57] S. Marinai, E. Marino, and G. Soda. Font adaptive word indexing of modern printed documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1187–1199, 2006.

[58] K. Marukawa, T. Hu, H. Fujisawa, and Y. Shima. Document retrieval tolerating character recognition errors–evaluation and application. *Pattern Recognition*, 30(7):1361–1371, 1997.

[59] M. Meshesha and C. V. Jawahar. Matching word images for content-based retrieval from printed document images. *IJDAR*, 11(1):29–38, 2008.

[60] M. Mitra and B. B. Chaudhuri. Information retrieval from documents: A survey. *Inf. Retr.*, 2(2/3):141–163, 2000.

[61] G. Nagy. Twenty years of document image analysis in PAMI. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):38–62, 2000.

[62] T. Nakai, K. Kise, and M. Iwamura. Camera-based document image retrieval as voting for partial signatures of projective invariants. In *ICDAR*, pages 379–383, 2005.

[63] P. Natarajan, E. MacRostie, and M. Decerbo. *Guide to OCR for Indic Scripts*, chapter PartI,9. Springer London, BBN Technologies, Cambridge, MA 02138, USA, 2010.

[64] U. Pal and B. B. Chaudhuri. Indian script character recognition: a survey. *Pattern Recognition*, 37(9):1887–1899, 2004.

[65] U. Pal and B. B. Chaudhuri. Indian script character recognition: a survey. *Pattern Recognition*, 37(9):1887–1899, 2004.

[66] D. Ramanan and S. Baker. Local distance functions: A taxonomy, new algorithms, and an evaluation. In *ICCV*, pages 301–308, 2009.

[67] R. G.-B. Ranb, A. Navot, and N. Tishby. Margin based feature selection - theory and algorithms. In *In International Conference on Machine Learning (ICML*, pages 43–50. ACM Press, 2004.

[68] V. Rasagna, A. Kumar, C. V. Jawahar, and R. Manmatha. Robust recognition of documents by fusing results of word clusters. In *ICDAR*, 2009.

[69] T. M. Rath, S. Kane, A. Lehman, E. Patridge, and R. Manmatha. Indexing for a digital library of george washington's manusripts: a study of word matching techniques. In *Center for Intelligent Information Retrieval*, 2000.

[70] T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *ICDAR*, pages 218–222, 2003.

[71] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *CVPR (2)*, pages 521–527, 2003.

[72] T. M. Rath and R. Manmatha. Word spotting for historical documents. *IJDAR*, 9(2-4):139–152, 2007.

[73] J. Rodriguez-Serrano, F. Perronnin, J. Llados, and G. Sanchez. A similarity measure between vector sequences with application to handwritten word image retrieval. *CVPR*, 2009.

[74] Y. Rui, T. S. Huang, and S. fu Chang. Image retrieval: Past, present, and future. In *Journal of Visual Communication and Image Representation*, pages 1–23, 1997.

[75] W. Saffady. *Introduction to Automation for Librarians*. 4th edition, 1999.

[76] L. Schomaker. Retrieval of handwritten lines in historical documents. In *ICDAR*, pages 594–598, 2007.

[77] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45:2673–2681, 1997.

[78] Y. Seiji and O. Takashi. Document retrieval using relevance feedback. *CICSJ Bull*, 21(1):32–36, 2004.

[79] S. Shalev-shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *In International Conference on Machine Learning (ICML*, pages 743–750. ACM Press, 2004.

[80] S. N. Srihari, S. Shetty, S. Chen, H. Srinivasan, C. Huang, G. Agam, and O. Frieder. Document image retrieval using signatures as queries. In *DIAL*, pages 198–203, 2006.

[81] C. Y. Suen, S. Mori, S. H. Kim, and C. H. Leung. Analysis and recognition of asian scripts - the state of the art. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pages 866–872, Washington, DC, USA, 2003.

[82] K. Taghva, J. Borsack, and A. Condit. Evaluation of model-based retrieval effectiveness with ocr text. *ACM Trans. Inf. Syst.*, 14(1):64–93, 1996.

[83] J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Computing*, 12(6):1247–1283, 2000.

[84] C. J. van Rijsbergen. *Information Retrieval*. Butterworth, 1979.

[85] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *NIPS*, pages 985–992, 2001.

[86] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005.

[87] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for aclass of lazy learning algorithms. *Artif. Intell. Rev.*, 11:273–314, February 1997.

[88] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 505–512, 2002.

[89] K. Zagoris, N. Papamarkos, and C. Chamzas. Web document image retrieval system based on word spotting. In *ICIP*, pages 477–480, 2006.

[90] Q. Zheng and T. Kanungo. Morphological degradation models and their use in document image restoration. In *ICIP (1)*, pages 193–196, 2001.

[91] L. Zhidong, S. Richard, N. Premkumar, B. Issam, and M. John. Advances in the bbn byblos ocr system. In *ICDAR*, pages 337–345, 1999.

[92] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia System*, 8(6):536–544, 2003.

[93] G. Zhu and D. S. Doermann. Logo matching for document image retrieval. In *ICDAR*, pages 606–610, 2009.

[94] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Signature detection and matching for document image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(11):2015 –2031, nov. 2009.

[95] G. Zhu, Y. Zheng, and D. S. Doermann. Signature-based document image retrieval. In *ECCV (3)*, pages 752–765, 2008.