International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Content Based Kannada Document Image Retrieval (CBKDIR)

Mr. Nithya. E,
Research Scholar, Dept of CSE
DayanandaSagar College of Engineering,
Bangalore – 78

Dr. Ramesh Babu D R,
Prof & HOD, Dept of CSE,
DayanandaSagar College of Engineering,
Bangalore – 78

Abstract – This paper presents a system for retrieval of content document from large kannada language document image collection. We can achieve the effective search and retrieval from a large collection of printed kannada documents images by matching image at word – level. For the representation of words, morphological operation is performed to get the height, width and coordination of each word in the document. On the given query image Fast Fourier Transformation is performed to determine the phase angle which is used to matching the words and retrieves the document. System level issues for retrieval are addressed in this.

Key Terms—
Fast Fourier Transformation, Inverse Fast Fourier Transformation, Morphological operation, Phase based image matching.

I. INTRODUCTION

The present growth of digitization of documents demands an immediate solution to enable the archived valuable materials searchable and usable by users in order to achieve its objective. High accuracy OCR systems are reported for English with excellent performance in presence of printing variations and document degradation. For Indian and many other oriental languages, OCR systems are not yet able to successfully recognize printed kannada document images of varying scripts, quality, size, style and font. Compared to European languages, kannada languages pose many additional challenges. Some of them are (i) Large number of vowels, consonants, and conjuncts, (ii) Most scripts spread over several zones, (iii) inflectional in nature and having complex character grapheme, (iv) lack of statistical analysis of most popular fonts and/or databases, (v) lack of standard test databases (ground truth data) of the kannada language\, add to the complexity of the design and implementation of a document image retrieval system..

A number of collections of historical prints, writings, manuscripts, news paper, large property ,official document and books exist in kannada languages that need search options in images. The document images of such collections cannot be recognized accurately. For Example the Famous Sanskrit poet and dramatist of India "Kalidasa" Books collection Such collections can be made available to large communities through electronic media. There is a need for easy and efficient access to such collections. The search procedures available for text domain can be applied, if these document images are converted into textual representations using recognizers. However, it is an infeasible solution due to the unavailability of efficient and robust OCR for kannada languages.

Addressing this problem, this paper proposes an efficient document image retrieval algorithm using phase based image matching – an image matching technique using the phase components in Fast Fourier Transformation which determines the phase angle of input image and query image that helps in matching word for the retrieval of document. This approach is faster as it does not match the image pixel by pixel.

II. LITERATURE SURVEY

In this chapter, we look at the literature of indexing and retrieval techniques used for search in large image databases. The topic of interest overlaps with databases, pattern recognition, content based image retrieval, digital libraries, document image processing and information retrieval.

A number of approaches have been proposed in recent years for efficient search and retrieval of document images. There are essentially two classes of techniques to search document image collections. The first approach is to convert the images into text and then apply a search engine.

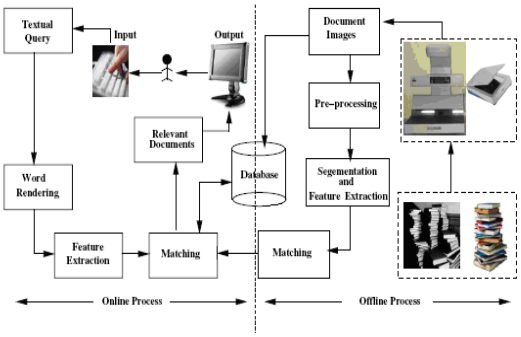
In recognition based search and retrieval techniques, the document images are passed through an optical character recognizer (OCR) to obtain text documents. The text documents are then processed by a text search engine to build the index. The text index makes the document retrieval efficient. An example is the gHMM approach of Chan et al. [2], suggested for printed and handwritten Arabic documents. It uses gHMMs with a bi-gram letter transition model, and kernel principal component analysis (KPCA) / linear discriminant analysis (LDA) for letter discrimination. In this approach segmentation and recognition go hand in hand. The words are modeled at letter level, where the likelihood of a word given a segment is used for discriminating words. The Byblos system [3] also uses a similar approach to recognize documents where a line is first segmented out and then divided in to image strips. Each line is then recognized using an HMM and a bi-gram letter transition model.

Taghva et al. [5] built a search engine for documents obtained after recognition of images. Searching is done based on the results of similarity calculation between the query words and the database words. The indexed terms are divided into two groups of correctly recognized and incorrectly

recognized words based on frequency calculations using a dictionary. Similar words are identified from the correct terms by applying mutual information measure. There have been attempts to retrieve complete documents (rather than searching words) by considering the information from word neighborhood (like n-grams [6]) to improve the search in presence of OCR errors [7].Word spotting is a method of searching and locating words in document images by treating a collection of documents as a collection of word images. The words are clustered and the clusters are annotated for enabling indexing and searching over the documents. It involves segmentation of each document into its corresponding lines and then into words. The word spotting approach [11] has been extended to searching queried words from printed document images of newspapers and books. Dynamic time warping (DTW) based word-spotting algorithm for indexing and retrieval of online documents is also reported in [12].

The remainder of the paper describes our current development effort in more detail. Section III describes the architecture of the research prototype we are developing. Section IV details the implementation evaluation procedures we are developing. Section V presents some experimental results. Section VI concludes the paper.

III. SYSTEM ARCHITECTURE



Fig(1) System Architecture

This system accepts a textual query from users. The textual query is first converted to an image by rendering, features are extracted from these images and then a search is carried out for retrieval of relevant documents. Results of the search are pages from document image collections containing the retrieved words sorted based on their relevance to the query. This work mainly aims at addressing some of the issues involved in effective and efficient retrieval in document images with effective representations of the content kannada word images.

III. IMPLEMENTATION

An efficient mechanism for retrieval of a Kannada word from a large document image collection is presented in this paper. This involves three phases: first phase includes pre processing, which is preparing the source image for searching the query word, second phase includes generating the query image from the query word, and third phase includes matching of images to find the query word in the source image

a. Preprocessing :

Preprocessing is the first phase. Consider the source images that are stored in the database, which is generally in form of RGB. The below figure(2) shows the following

Fig(2) The source image stored in the database

The source image, which is in the RGB form, is converted to the grayscale image. "**rgb2gray.m"** is the built in function in Matlab which converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance. "**rgb2gray.m**"converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components: 0.2989 * R + 0.5870 * G + 0.1140 * B. The figure(3) below shows the conversion

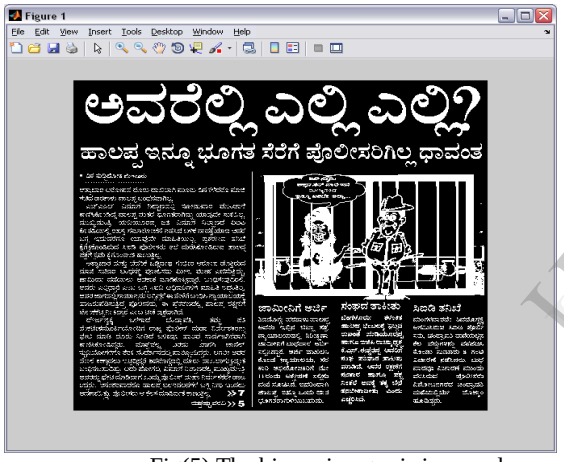
Fig(3) The source image converted from RGB to Grayscale

Then, this grey scale image is, in turn, converted to the binary image, i.e. image will be in 0's and 1's form, with 0 representing black and 1 representing white. "**im2bw.m"** is the built in function in Matlab which converts image to binary image, based on threshold. **BW = im2bw (I, level)** converts the grayscale image I to a binary image.

The output image BW replaces all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black). You specify level in the range [0,1], regardless of the class of the input image. If you do not specify level, im2bw uses the value 0.5. The fig(4) below allow shows the conversion.

Fig(4) The source image converted from grayscale to binary image

The source image is inversed which makes the background black and foreground white, which will help in segmentation stage. This is done by inversing the variable which contains the binary image. For example if **bw** contains the binary image then **bw=~bw;** will inverse the colour. The fig(5) below shows the conversion.

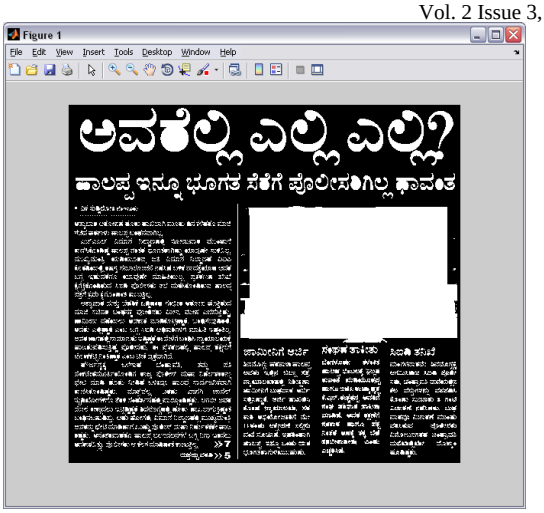


Fig(5) The binary images is inversed

Then, fill the image regions and holes to find any picture in the image. **imfill** is the built in function used in Matlab to fill the regions and holes.

For example **`BW1 = imfill (BW, 'holes')`** fills holes in the binary image `BW`. A hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image. The below figure(6) show the filling of holes.

Then the big areas are removed, which might be, say, a photograph. **bwareaopen** is the built in function used in Matlab to morphologically open binary image (remove small objects). For example **BW2 = bwareaopen(BW,P)** removes from a binary image all connected components (objects) that have fewer than P pixels, producing another binary image, BW2. The default connectivity is 8 for two dimensions, 26 for three dimensions, and **conndef (ndims (BW),'maximal')** for higher dimensions. The fig(6.6) below show the removal of photographs from the document.



Fig(6) The inversed image where the image regions and holes are filled

Fig(7) The big areas such as photographs are removed from the image document

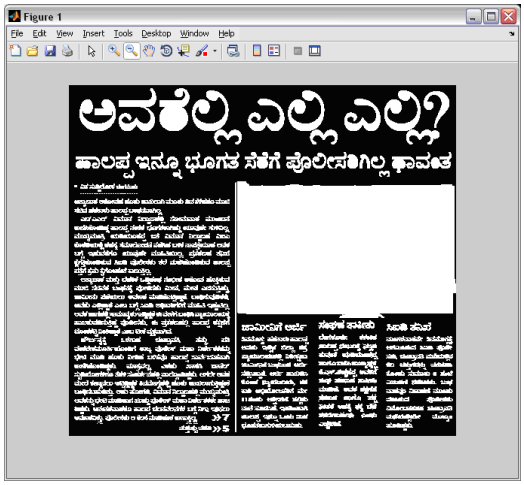
This whole process is done to perform morphological operation. Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are **dilation** and **erosion**. Dilation adds pixels to the boundaries of objects in an image. Morphological operation is performed to initiate the dilation, which helps in differentiating two words delimited by a space. imdilate is the built in function used in Matlab to dilate the document. For example **IM2 = imdilate(IM, SE)** dilates the grayscale, binary, or packed binary image IM, returning the dilated image, IM2. The argument SE is a structuring element object, or array of structuring element objects, returned by the strel function.

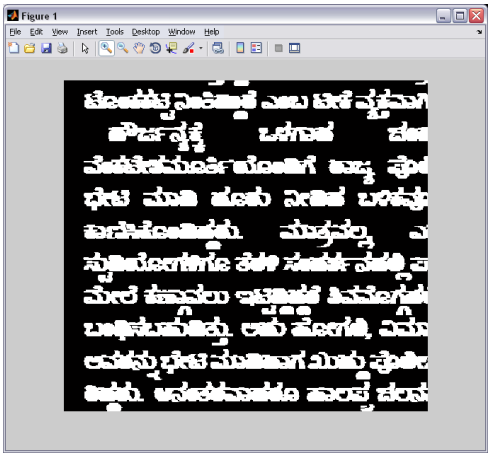
If IM is logical and the structuring element is flat, imdilate performs binary dilation; otherwise, it performs grayscale dilation. If SE is an array of structuring element objects, imdilate performs multiple dilations of the input image, using each structuring element in SE in succession.

The dilation can be done in different shapes like line, diamond, octagon, square etc. Before performing the dilation we need to choose the shape in which the dilation has to be done. strel function can be used for this purpose. For example if the dilation has to be done in line then **SE = strel('line', LEN, DEG)** creates a flat, linear structuring element, where LEN specifies the length, and DEG specifies the angle (in degrees) of the line, as measured in a counterclockwise direction from the horizontal axis. LEN is approximately the distance between the centers of the structuring element members at opposite ends of the line.

The figure (8) and figure(9) below shows the morphological operation done on the document. The morphological operation used on the document is horizontal line dilation. The process helps to segment the words easily in the document.



Fig(8) Image after horizontal line dilation



Fig(9) Zoomed image of horizontal line dilation

Then at last , record the coordinates, height and width of each word in image document. **"bwlabel.m**" is the built in function used in Matlab to label word areas in binary image which gives the coordinates of all the words in the image document which speeds up the matching stage.

b. Preprocessing for query image:

Read the query text. Read the image from the database folder as per the query text. Perform two preprocessing methods on the image. First preprocessing is **rgb2gray I=rgb2gray(I);**and then gray image to binay image **bw=im2bw(I,0.7);**. Later inverse it so that we can get the proper written document **bw=~bw;.**

Normalize the image to one size in order to concatenate all the query images. **Find.m** function is used which will give the nonzero terms in an image. Then the minimum row and column value of the nonzero term is found out and the point will be aligned. In some special cases such as ಹ , ಖ we have to put the condition on the minimum value of row, so that we can align the letter to same size .

c. Phase based word image matching :

Matching the word is the 3rd phase. This includes matching of images to find the query word in the source image. This phase has input as two images, source image and query image, which are converted into binary form. First, Fast Fourier transform is performed on both the images and phase angle of both the images are determined. Then, subtract the phase angle of first image with that of the second image. Inverse Fast Fourier transform is performed on thus obtained phase difference. If the highest value of IFFT result is more than the threshold then words are matching, else the words are not matching.

In this section, the principle of phase-based image matching using the limited Phase-Only Correlation (POC) function (which is sometimes called the "phase-correlation function")is explained . Consider two $N1 \times N2$ images, $f(n1, n2)$ and $g(n1, n2)$, where we assume that the index ranges are $n1 = -M1 \cdots M1$ $(M1 > 0)$ and $n2 = -M2 \cdots M2$ $(M2 > 0)$ for mathematical simplicity, and hence $N1 = 2M1 + 1$ and $N2 = 2M2 + 1$. Let $F(k1, k2)$ and $G(k1, k2)$ denote the Fast Fourier Transform(FFT) of the two images. $F(k1, k2)$ is given by

$$F(k_1, k_2) = \sum_{n1,n2} f(n_1, n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2}$$

$$= A_F(k_1, k_2) e^{j\theta_F(k_1, k_2)}$$

where $k1 = -M1 \cdots M1$, $k2 = -M2 \cdots M2$, $W_{N_1} = e^{-j\frac{2\pi}{N_1}}$, $W_{N_2} = e^{-j\frac{2\pi}{N_2}}$ and $\sum_{n_1,n_2}$ denotes $\sum_{n_1=-M_1}^{M_1} \sum_{n_2=-M_2}^{M_2}$ $AF(k1, k2)$ is amplitude and $\theta_F(k1, k2)$ is phase. $G(k1, k2)$ is defined in the same way. The cross-phase spectrum $R_{FG}(k1, k2)$ is given by

$$R_{FG}(k_1,k_2)=\frac{F(k_1,k_2)\overline{G(k_1,k_2)}}{\left|F(k_1,k_2)\overline{G(k_1,k_2)}\right|}$$

$$= e^{j\theta(k_1,k_2)}$$

where $G(k1, k2)$ is the complex conjugate of $G(k1, k2)$ and $\theta(k1, k2)$ denotes the phase difference $\theta_F(k1, k2)-\theta_G(k1, k2)$. The POC function $r_{fg}(n1, n2)$ is the Inverse Fast Fourier Transform (IFFT) of $R_{FG}(k1, k2)$ and is given by

$$r_{fg}(n_1,n_2)=\frac{1}{N_1N_2}\sum_{k_1,k_2}R_{FG}(k_1,k_2)W_{N_1}^{-k_1n_1}W_{N_2}^{-k_2n_2}$$

where $\sum_{k_1,k_2}$ denotes $\sum_{k_1=-M_1}^{M_1}\sum_{k_2=-M_2}^{M_2}$ When two images are similar, their POC function gives a distinct sharp peak. When two images are not similar, the peak drops significantly. The height of the peak gives a good similarity measure for image matching, and the location of the peak shows the translational displacement between the images.

i) Fast Fourier Transform (FFT):

Fast Fourier Transform (FFT) algorithms have computational complexity O(n log n) instead of $O(n^2)$. If n is a power of 2, a one-dimensional FFT of length n requires less than 3n $\log_2$ n floating-point operations (times a proportionality constant). For n = 220, that is a factor of almost 35,000 faster than $2n^2$.

When using FFT algorithms, a distinction is made between the window length and the transform length. The window length is the length of the input data vector. It is determined by, for example, the size of an external buffer. The transform length is the length of the output, the computed DFT. An FFT algorithm pads or chops the input to achieve the desired transform length.

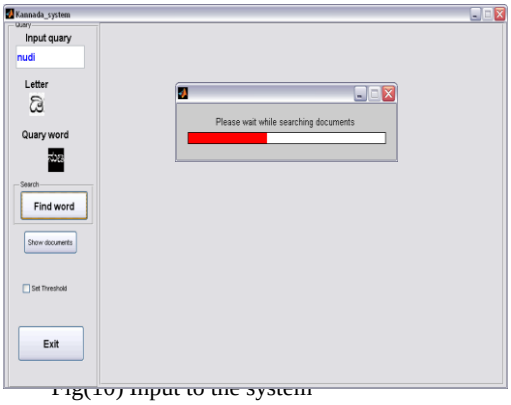
The execution time of an FFT algorithm depends on the transform length. It is fastest when the transform length is a power of two, and almost as fast when the transform length has only small prime factors. It is typically slower for transform lengths that are prime or have large prime factors. Time differences, however, are reduced to insignificance by modern FFT algorithms such as those used in MATLAB. Adjusting the transform length for efficiency is usually unnecessary in practice.

ii) Inverse Fast Fourier Transform (IFFT):

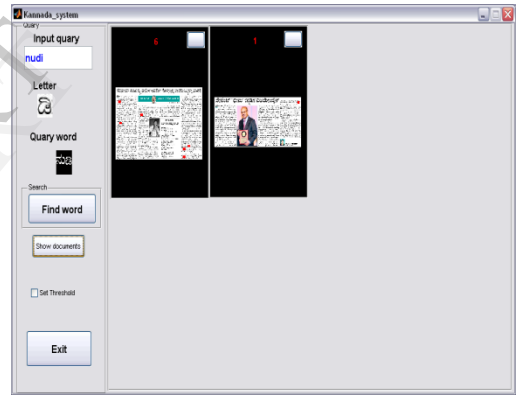
IFFT(X) is the same as the algorithm for FFT(X), except for a sign change and a scale factor of n = length(X). As for FFT, the execution time for IFFT depends on the length of the transform. It is fastest for powers of two. It is almost as fast for lengths that have only small prime factors. It is typically several times slower for lengths that are prime or which have large prime factors.

IV. EXPERIMENTAL RESULTS

Figure (10) shows the input to the system and figure (11) shows the output. Since we are using database approach for the character recognition, in this approach for each character we need to have details like Character name, Character BMP image. This takes lot of space as well as lot of computation involved in recognizing the character.



Fig(10) Input to the system



Fig(11) Result of search

V. CONCLUSION

In this paper we have presented content based kannada Docment image retrieval in large kannada document image collections. The matching technique is based on phase - based image matching, for search in large collections of document word images is applied to obtain good performance. The approaches used for word spotting so far, dynamic time warping and/or nearest neighbor search tend to be slow for large collection of books. Direct matching of pixels in images is inefficient due to the complexity of matching and thus impractical for large databases. This problem is solved by directly storing word image representations.

Some of the possible directions in which the future work can be carried out are as below. The effect of combination of different fonts in a single collection can be one possible direction for exploring the feasibility of the proposed

technique and improving it. Multi-lingual document images are not handled in the proposed technique.

REFERENCE:

[1] Koichi Ito, Ayumi Morita, Takafumi Aoki, Tatsuo Higuchi, Hiroshi Nakajima, and Koji Kobayashi, A Fingerprint Recognition Algorithm Using Phase- Based Image Matching for Low-Quality Fingerprints

[2] Ashwin T V 2000 *A font and size independent OCR for printed Kannada using SVM*. M E Project Report, Dept. Electrical Engg., Indian Institute of Science, Bangalore

[3] A. Balasubramanian, Million Meshesha, and C.V. Jawahar Retrieval from Document Image Collections Centre for Visual Information Technology, International Institute of Information Technology, Hyderabad - 500 032, India

[4] Ashwin T V 2000 A *font and size independent OCR for printed Kannada using SVM* M E Project Report, Dept. Electrical Engg., Indian Institute of Science, Bangalore

[5] Bansal V, Sinha R M K 1999 On how to describe shapes of Devanagari characters and use them for recognition. In *Proc. Fifth Int. Conf. on Document Analysis and Recognition*, Bangalore (IEEE Computer Society Press) pp 410–413

[6] H.S. Baird: Background Structure in Document Images. Bunke H and Wang P S P and Baird, H S (Eds.), Document Image Analysis, World Scientific, Singapore, pp. 17 – 34, 1994

[7] T M Breuel: Two Geometric Algorithms for Layout Analysis. In DAS '02: Proceedings of the 5th International Workshop on Document Analysis Systems V, Springer – Verlag, London, UK, pp. 188 – 199, 2002.

[8] Bansal V, Sinha R M K 1999 On how to describe shapes of Devanagari characters and use them for recognition. In Proc. Fifth Int. Conf. on Document Analysis and Recognition, Bangalore (IEEE Computer Society Press) pp 410 – 413

[9] R Sanjeev Kunte and R D Sudhaker " A simple and efficient optical character recognition system for basic symbols in printer kannada text"

[10] B. Vijay Kumar and A G Ramakrishnan, Department of Electrical Engineering Machine Recognition of Printed Kannada Text Indian Institute of Science, Bangalore .

[11] S Basavaraj Patil and N V Subbareddy, Neural network based system for script identification in Indian documents, Kuvempu University Research Centre, Department of Computer Science and Engineering, University B D T College of Engineering, Davanagere

[12] B. VijayKumar, A G Ramakrishnan Dept of Electrical Engg. Radial Basis Function and subspace Approach for printed Kannada text Recognition, India Institute of Science Bangalore, India – 560012.