IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. VI (Jul.-Aug. 2016), PP 39-47
www.iosrjournals.org

Handwritten Kannada Document Image Processing using Optical Character Recognition

Mayur M Patil, Akkamahadevi R Hanni
Department of Computer Science, B V Bhoomaraddi College of Engineering and Technology, Hubli, Karnataka, India

Abstract: *The objective of Optical Character Recognition (OCR) is automatic reading of optically sensed document text materials to translate human-readable characters to machine- readable codes. In Optical Character Recognition, the text lines in a document must be segmented properly before recognition. English Character Recognition (CR) has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. But same is not the case for Indian languages which are complicated in terms of structure and computations. This is the motivation behind choosing OCR for Kannada language. A KSRTC bus pass application form written in Kannada is chosen for processing and recognition. The OCR system is devised to first segment the whole document into text lines, then to words and then to individual characters. These characters are then used to extract the necessary features and recognize those characters and classify them.*

Keywords: *Optical Character Recognition (OCR), Character Recognition (CR).*

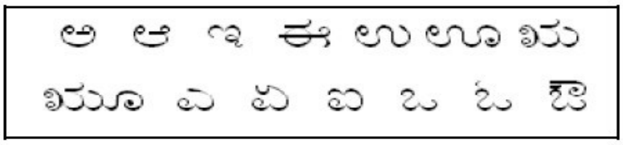
I. Introduction

Optical character recognition has many different practical applications. The main areas where OCR has been of importance are text entry(office automation), data entry (banking environment) and process automation (mail sorting).The present state of the art in OCR has moved from primitive schemes for limited character sets, to the application of more sophisticated techniques for omnifont and handprint recognition. The main problems in OCR usually lie in the segmentation of degraded symbols which are joined or fragmented. In spite of the great number of algorithms that have been developed for character recognition, the problem is not yet solved satisfactory, especially not in the cases when there are no strict limitations on the handwriting or quality of print. Given the testing sample which is a scanned handwritten document of BMTC bus pass application form, each character needs to be recognized in the form. Choosing proper and apt preprocessing steps and segmentation algorithms plays a vital role in the process as this being the initial step. Efficient feature extraction and classification methods should be used to maintain good performance and accuracy of results.

Samples of handwritten numerals and characters written by different people should be collected and used to create the training database. The BMTC bus pass application forms filled by different people will be taken as an input to the OCR system.

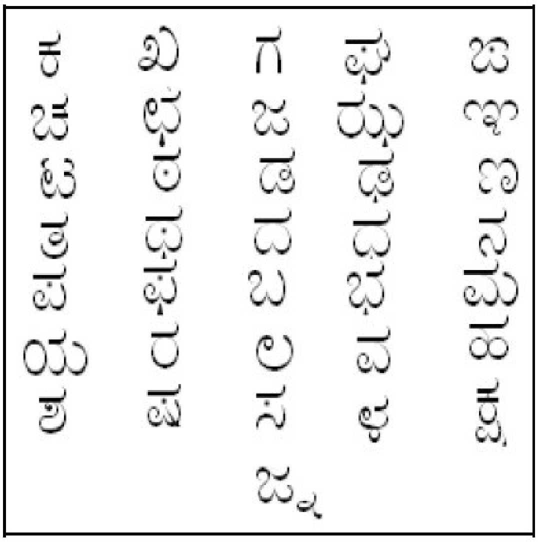
A. The characteristic of Kannada script

In this section, we will briefly describe some of the main characteristics of Kannada script to point out the main difficulties for segmenting.
Kannada is one of the South Indian languages which has 16 vowels and 35 consonants.



Vowels of Kannada Script

A Character can be one of the following,
i. A standalone vowel or a consonant
ii. A consonant modified by a vowel.
iii. A consonant modified by one or more consonants and a vowel.



Consonants of Kannada Script

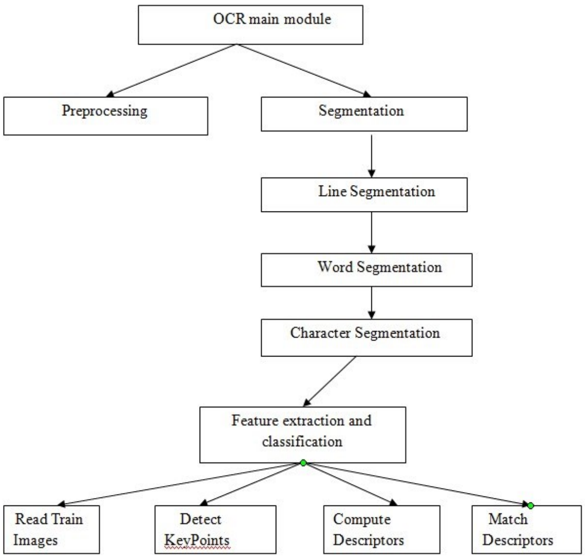


Fig 1. Different Modules and the flow of control in the OCR system

II. Preprocessing

This step is done to enhance the document by removing noise and other distortions in the written material. This step includes **skew detection and removal** and **noise removal**. When a document is fed to the optical sensor either (scanner) mechanically or by a human operator to get the digital image, a few degrees of skew (tilt) is unavoidable. Skew angle is the angle that the text lines in the digital image makes with the horizontal direction. Skew estimation and correction are important preprocessing steps of line and word segmentation approaches. Skew correction can be achieved by

(i) Estimating the skew angle, and
(ii) Rotating the image by the skew angle in the opposite direction.

There are three main reasons for removing the skew.

The first is appearance. Anything more than about 0.25 degrees (0.004radian) is quite noticeable. The second is that it is important to remove skew if any analysis is to be done on the page. The presence of skew, and particularly more than 0.01radians, complicates the analysis of page elements such as text columns. The third is that the performance of symbol- based compression on multipage documents is badly degraded by random skew of 0.01 radian or more, because the same characters are placed in different equivalence classes due to skew.

The methods for determining skew [1] are:

- Use all the pixels
- Use projection profiles
- Use the variance of the projection profiles
- Use the variance of the projection profile derivative
- Using linear and binary search

A projection profile based method [2] is being implemented. A straightforward solution to determining the skew angle of a document image uses a horizontal projection profile. This is a one-dimensional array with a number of locations equal to the number of rows in an image. Each location in the projection profile stores a count of the number of black pixels in the corresponding row of the image. This histogram has the maximum amplitude and frequency when the text in the image is skewed at zero degrees since the number of co-linear black pixels is maximized in this condition. The peaks in the profile calculated from the de skewed image (Fig.4) shown in (Fig6) are taller and more closely spaced than those computed from the skewed image (Fig.3) shown in (Fig5). This characteristic has been used in several algorithms. One way of doing this is to rotate the input image through a range of angles and calculate the projection profile for each one. Then features extracted from each projection profile are compared to determine which one is more peaked.

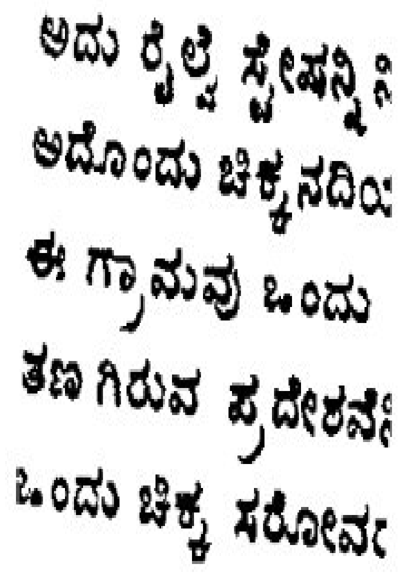


Fig 2. Skewed Image

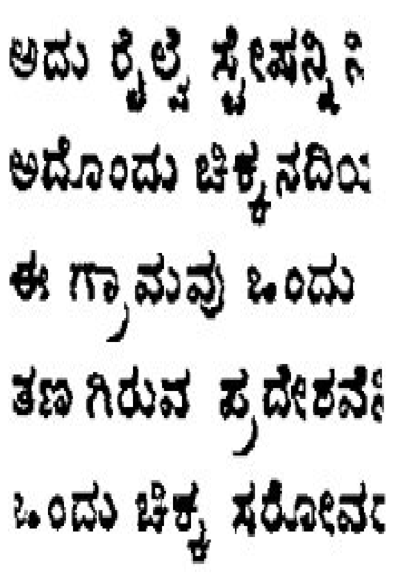


Fig 3. De-skewed Image

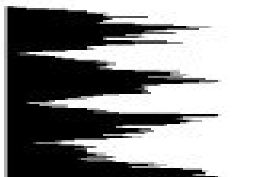


Fig4. Horizontal projection of skewed image

Fig5. Horizontal projection of de-skewed image

The noise removal is done using a median filter. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing (for example, edge detection on an image) .Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise [12]. The median filter [11] works as follows: Neighborhood averaging can suppress isolated out-of-range noise, but the side effect is that it also blurs sudden changes (corresponding to high spatial frequencies) such as sharp edges. The median filter is an effective method that can suppress isolated noise without blurring sharp edges. Specifically, the median filter replaces a pixel by the median of all pixels in the neighborhood:

$$y[m,n] = median\{x[i,j],(i,j) \in w\}$$

Where w represents a neighborhood centered around location (m, n) in the image. Programming issues: Sorting is necessary for finding the median of a set of values. There exist various sorting algorithms with complexity of O (n*log(n)).

Since only the handwritten text is being processed it needs to be extracted from the document containing both the handwritten and the printed text. A template which in the context of the project refers to an empty application form should be used to extract only the handwritten text.

III. Segmentation

Segmentation is an important task of any Optical Character Recognition (OCR) system. It separates the image text documents into lines, words and characters. Segmentation of handwritten text of some Indian languages like Kannada, Telugu, Assamese is difficult when compared with Latin based languages because of its structural complexity and increased character set. It contains vowels, consonants and compound characters. Some of the characters may overlap together. Despite several successful works all over the world, development of such tools in specific languages is still an ongoing process especially in the Indian context.

For an optical character recognition (OCR) system, segmentation phase is an important phase and accuracy of any OCR heavily depends upon segmentation phase. Incorrect segmentation leads to incorrect recognition. Segmentation phase includes line, word and character segmentation. Before word and character segmentation, line segmentation is performed to find the number of lines and boundaries of each line in any input document image. Incorrect line segmentation may result in decrease in recognition accuracy.

A. Challenges in segmentation:

In the recent past, the number of document images available for Indian languages has grown drastically with the establishment of Digital Library of India. The digital library documents originate from a variety of sources, and vary considerably in their structure, script, font, size, quality, etc. Text line extraction from unconstrained handwritten documents is a challenge because the text lines are often skewed and the space between lines is not obvious. The complexity involved in the segmentation of characters in the uneven spacing between text lines and adjacent characters. Of these, the variations in the structure of the script are the most taxing to any segmentation algorithm. Some of them are discussed in [5] are mentioned below. The complexity of the scripts lies in the spatial distribution of the connected components. Unlike English, most characters in Indian scripts are made up of more than one connected component. These connected components do not form meaningful characters by themselves, but when grouped together, form different characters in the alphabet. The components of a character can be classified into:

Main Component: It is either a vowel, a consonant or a truncated consonant. The main components of characters within a line are nearly collinear.

Consonant Modifier: In the above scripts, a character could be composed of two consonants, the main component and a consonant modifier or half consonant. Spatially, the consonant modifier could be to the left, right or bottom of the main component, and hence lie within a line, or below it.

Vowel Modifier: A character also can have a vowel modifier, which modifies the consonant. When the vowel modifier does not touch the main component, it forms separate component, which lies above the main component. Due to variations in spatial distribution of the components within a line, the line structure is non-uniform. This is the primary reason for the failure of many traditional segmentation algorithms. Due to the positional variation of a modifier component, the task of assigning it to a line above it or below it is ambiguous. Heuristics such as assigning a component to its nearest line might fail because the distances between the components vary depending on the font style, font size and typeset. Variations in scanned books are also introduced due to the change in writing style of certain character overtime, which need to be taken into account while segmenting a document. Most of the old books are typeset by human and not machine and hence it is difficult to specify a consistent distance between the components. Curved and non-parallel text lines in handwritten documents also make the segmentation and recognition challenging.

B. Line Segmentation:

The simplest and most widely used method to segment the lines is to use the inter-line gap in horizontal projection as line boundaries. This technique may not work well on many documents in Indian scripts. It results in strips containing multiple horizontally overlapping lines leading to under segmentation and strips containing components of a line leading to over segmentation [3]. A variation and improvement in this approach leads to better results.

A brief outline of the static segmentation method is as below:

- Row wise dissection: (horizontal projection)
- *C*alculate row-wise pixel sums of the inputs
- *O*btain the row-wise projection of the inverted inputs
- *F*ind the minimum of the projections
- *A* pair of closely lying minimum points defines one segmentation boundary.

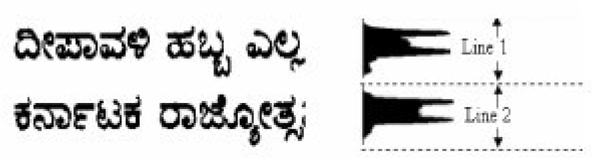


Fig6. Line segmentation using horizontal projection profile

Resolving the problems of overlapping and touching component [4]

In a text-page, either overlapping or touching or both the problems of overlapping and touching may occur in many positions of two consecutive text lines in the text-page. Therefore, the problem of text line segmentation cannot be completed without treating the overlapping and touching cases. In the present technique subsequent to getting separating lines, it should be checked whether each separating line passes through the white gap between two consecutive lines or it crosses some components of text lines. To do so, each separating line is traced from left to right and as soon as the separating line passes through black pixels of a component in the text-page, a problem of touching/overlapping happens. If a separating line does not pass through any black pixel of any component then neither touching nor overlapping occurs in that separating line. In order to check whether an overlapping or a touching has occurred, the height of the component having intersection with separating line is examined. If the height of the component, which having intersection with separating line, is greater than average height of components present in the input text-page, the component is judged as a touching component. Then the image is divided horizontally again for further processing.

Some other line segmentation algorithms are text line segmentation by clustering with distance metric learning [6], Nearest Neighbour Clustering approach for line segmentation[8], three stage line segmentation based on fringe map[9], algorithm using bi-variate Gaussian densities.[7]

C. Word segmentation:

In word segmentation method, a text line is taken as an input. After a text line is segmented, it is scanned vertically.

- Column wise dissection: (vertical projection)
- *C*alculate the sum of pixels column-wise.
- *O*btain the column-wise projection of the inverted sub-images.

- *F*ind the minimum points from these projections
- *A* pair of consecutive minimum points defines one segmentation boundary.

Since the words do not touch each other word segmentation will not pose issues.

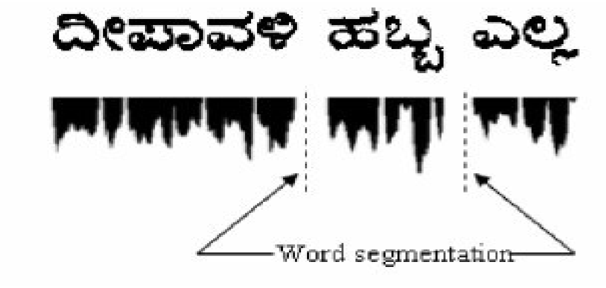


Fig7. Word Segmentation

D. Character Segmentation:

The segmentation and merge approach for segmentation of a Kannada word. In this method the words are vertically segmented into three zones. This segmentation is achieved by analyzing the HPP of a word. Separating the top zone from the bottom zone is easier as the consonant conjuncts are usually disconnected from the base consonant.

The top zone is first over segmented by extracting points in the vertical projection showing valleys in the histogram exceeding a fixed threshold. The threshold is kept low so that a large number of segments are obtained. This segmentation does not give consistent segments, these segments are merged using heuristic merging algorithm and recognition based algorithm. Then the bottom zone is analyzed to detect the consonant conjuncts. The accuracy of the classifier will affect the accuracy of segmentation.

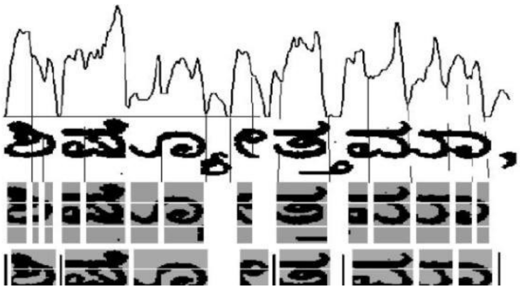


Fig8. Character Segmentation

E. Feature Extraction

Feature point descriptors are now at the core of many Computer Vision technologies, such as object recognition, 3D reconstruction, image retrieval, and camera localization. Since applications of these technologies have to handle more data or to run on mobile devices with limited computational resources, there is a growing need for local descriptors that are fast to compute, fast to match and memory efficient.

One way to speed up matching and reduce memory consumption is to work with short descriptors. They can be obtained by applying dimensionality reduction, such as PCA or LDA, to an original descriptor such as SIFT or SURF.

In this feature extraction we match the descriptors detected on one image to descriptors detected in the image set. So we will have one query image and several train images. For each key point descriptor of query image the one nearest train descriptor is found in the entire collection of train images. To visualize the result of matching we save the images, each of which combines query and train image with matches between them. Match is drawn as line between corresponding points. Count of all matches is equal to the count of query key points, so we have the same count of lines in all set of result images. Feature Detector [13] is used to detect the key points in an image or image set. SURF based Feature Detector method is used. There are different Feature Detector types like
"FAST"– Fast Feature Detector
"STAR"– Star Feature Detector
"SIFT"– Sift Feature Detector
"SURF"– Surf Feature Detector
"ORB"– Orb Feature Detector
"MSER"– Mser Feature Detector
"GFTT"– Good Features to Track Detector
"HARRIS"– Good Features to Track Detector with Harris detector enabled.

SURF Feature Detector: SURF (Speeded Up Robust Feature) is a robust image detector & descriptor. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images. It uses an integer approximation to the determinant of Hessian blob detector, which can be computed extremely quickly with an integral image (3 integer operations). For features, it uses the sum of the Haar wavelet response around the point of interest. SURF is fast and has good performance than SIFT, but it is not stable to rotation and illumination changes.

Key Point Detection:
-First we build the Integral Image Using Pyramid of Filter (not image) to approximate Laplace of Gaussian (LoG), supposedly run faster than SIFT.
-Filters will span several octaves and with fixed number of scales in each, similar to SIFT. This type of process is called
Scale-Space Analysis.
-Integral image helps to keep the running speed constant as it is insensitive to increasing filter sizes.
-It Uses discrete Box Filter to calculate Hessian determinant and Box size is multiples of its current scale.
-All filters run on the original image instead of iterative like SIFT, allowing parallel execution.
-Finally the Feature points are maxima of the determinants in the adjacent scale and points (3x3x3), similar to SIFT.
Orientation Assignment:
-The responses from 2 first-order Haar wavelet filters (1,-1), in dx and dy orientations, are collected on each feature point. The responses are put on a 2Dplaneasvectors[dx, dy] .
-Again, the Integral Image would help with this box filter.
-An orientation-window of 60-degree-angle is slided around the origin at x-y plane. Each angle will have a corresponding sum of magnitudes for every vectors inside that window.
-The dominant orientation would be the angle of the largest sum.
Descriptor Extraction:
-The descriptor is 64-element vector, half the size of SIFT. Each represents the intensity property of a 4x4 square within the "interest region". The intensity property is 4-element tuple [Sum(dx),Sum(abs(dx)), Sum(dy), Sum(abs(dy))].The Sum is calculated from the Haar wavelet response of a 5x5 sample window.
-4x4x4= 64
-Since there are more than 14x4 square at each key-point location. We can say that the size of the region is related to the associated scale.
Matching:
The sign (+ive/-ive) of the Trace of Laplacian (Hessian
Matrix) is used in matching phase, speeding up the process.

IV. Classification and recognition

The Descriptor Matcher used is FLANN. It is an Abstract base class for matching key point descriptors. It has two groups of match methods: for matching descriptors of an image with another image or with an image set. There are different types of Descriptor Matcher like 'brute-force','flann' etc. There are different methods to match descriptors from an image pair or an image set like match, knn Match, radius Match Descriptor Matcher::match finds the best match for each descriptor from a query set. Descriptor Matcher::knnMatch finds the k best matches for each descriptor from a query set. Descriptor Matcher::radiusMatch for each query descriptor, finds the training descriptors not farther than the specified distance.

FLANN provides a library of feature matching methods. It can provide automatic selection of index tree and parameter based on the user's optimization preference on a particular data-set. They are chosen according to the user's preferences on the importance between build time, search time and memory footprint of the index trees. The type of index trees are KD, randomized KD and Hierachical K-Means.

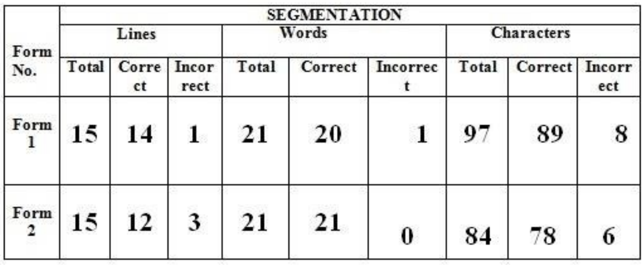
FLANN uses the Hierachical K-means Tree for generic feature matching. Nearest neighbors are discovered by choosing to examine the branch-not-taken nodes along the way. FLANN uses a priority-queue (Best-Bin-First) to do ANN from Hierachical K-Means Tree.

A. Algorithms Used

Algorithms used in FLANN

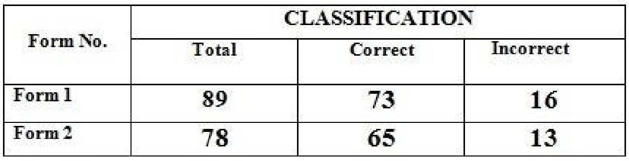
- K-D Tree
- Randomized K-D Tree
- ANN Search
- On K-D Tree
- On RKD Tree
- On Hierarchical K-Means Tree

V. Results



Form No.	SEGMENTATION								
	Lines			Words			Characters		
	Total	Correct	Incorrect	Total	Correct	Incorrect	Total	Correct	Incorrect
Form 1	15	14	1	21	20	1	97	89	8
Form 2	15	12	3	21	21	0	84	78	6

TABLEI. The segmentation results obtained in different forms



Form No.	CLASSIFICATION		
	Total	Correct	Incorrect
Form 1	89	73	16
Form 2	78	65	13

TABLEII. The classification results obtained in different forms

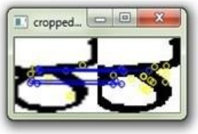


Figure 9 correctly classified alphabet

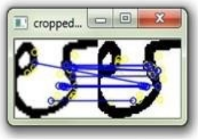


Figure 10 correctly classified numeral

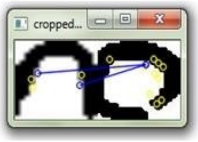


Figure 11 Incorrectly classified numeral

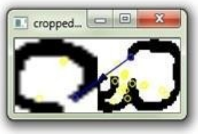


Figure 12 Incorrectly classified character

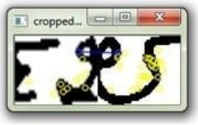


Figure 13 Incorrectly segmented

VI. Conclusion and future scope

Developing an OCR for a handwritten BMTC bus pass form is quite challenging and prone to errors due to structural complexity and increased character set of Kannada language. An attempt is made in this direction and the recognition of characters is done. Better skew detection and noise removal techniques can be used to enhance the Preprocessing and Segmentation phases. Efficient feature extraction and classification methods are used to get good performance and accuracy of results. This can be further enhanced to convert the recognized characters to electronic form. This can be further extended for other form based applications like forms used in handling deposits and withdrawals in banks, educational institutions, applications in government offices, etc. The results are found satisfactory for the algorithms used for the current system. The system may be required to be modified slightly if used for other form based applications processing.

References

[1]. Dan Bloomberg, ”Analysis of Document Skew”,Leptonica2002 [2]JONATHAN J.HULL,” Document Image Skew Detection: Survey And Annotated Bibliography”, Document Analysis Systems

[2]. J.J.Hull, S.L.Taylor, Eds., World Scientific, pp.40-64,1998.

[3]. M.K.Jindal, R.K.Sharma, G.S.Lehal,” Segmentation of Horizontally Overlapping Lines in Printed Indian Scripts”, International Journal of Computational Intelligence Research.

[4]. ISSN 0973-1873Vol.3,No.4(2007), pp.277–286 Nallapareddy Priyanka, Srikanta Pal, Ranju Mandal,” Line and Word Segmentation Approach for Printed Documents”, IJCA Special Issue on “Recent Trends in Image Processing and Pattern Recognition” RTIPPR, 2010.

[5]. K.S.SeshKumar, A.M.Namboodiri, and C.V.Jawahar,” Learning Segmentation of Documents with Complex Scripts”, ICVGIP 2006,LNCS 4338, pp. 749–760,2006

[6]. Fei Yin, Cheng-Lin Liu,” Handwritten Chinese text line segmentation by clustering with distance metric learning”, Pattern Recognition 42(2009)3146 –3157

[7]. Manivannan Arivazhagan, Harish Srinivasanand Sargur Srihari,” A Statistical approach to line segmentation in handwritten Documents” ,Center of Excellence for Document Analysis and Recognition(CEDAR)

[8]. K. Srikanta Murthy, G.Hemantha Kumar, P.Shivakumar, P.R. Ranganath,” Nearest Neighbor Clustering approach for line and character segmentation in epigraphical scripts”

[9]. Vijaya Kumar Koppula , AtulNegi,” Using Fringe Maps for Text Line Segmentation in Printed or Handwritten Document Images ”, Second Vaagdevi International Conference on Information Technology for Real World Problems 2010

[10]. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool,” Speeded-Up Robust Features (SURF)”, Computer Vision and Image Understanding 110(2008)346–359

[11]. http://fourier.eng.hmc.edu/e161/lectures/smooth_sharpen/node3.html

[12]. http://en.wikipedia.org/wiki/Median_filter

[13]. http://experienceopencv.blogspot.com/2011_01_01_archive.html