Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPROVING OCR BY EFFECTIVE PRE-PROCESSING AND SEGMENTATION FOR DEVANAGIRI SCRIPT:A QUANTIFIED STUDY

[1] Dr. DEEPA GUPTA, [2] LEEMA MADHU NAIR

[1] Department of Mathematics, Amrita School of Engineering, Bangalore campus, Amrita Vishwa VidyaPeetham, Karnataka,INDIA

[2]Department of Computer Science, Amrita School of Engineering, Bangalore campus, Amrita Vishwa VidyaPeetham, Karnataka,INDIA

E-mail: [1]deepagupta.verma@gmail.com, [2]leemamadhu@gmail.com

ABSTRACT

Optical Character Recognition (OCR) system aims to convert optically scanned text image to a machine editable text form. Multiple approaches to preprocessing and segmentation exist for various scripts. However, only a restricted combination of the same has been experimented on Devanagari script. This paper proposes a study which aims to explore and bring out an alternative and efficient strategy of pre-processing and segmentation in handling OCR for Devanagari scripts. Efficiency evaluation of the proposed alternative has been undertaken by subjecting it to documents with varying degree of noise severity and border artifacts. The experimental results confirm our proposition to be superior approach over other conventional methodologies to OCR system implementation for Devanagari scripts. Also described is detailed approach to conventional pre-processing involved in initial stage of OCR, including noise removal techniques, along with the other conventional approaches to segmentation. The proposed alternative has been deployed to reach character and top character segmentation level.

Keywords: *Optical Character Recognition(OCR), Pre-processing, segmentation, Morphological operators, Connected component, Projection profile, Noise removal, Devanagari.*

1. INTRODUCTION

The digital era has made paper documentation an extinct process for everything today is done with the help of computers. Today, it is important that all such documentations be stored as digital data since digital memory is very cheap in comparison to real time storage spaces. Besides, such storage is less susceptible to damage, loss or theft. Optical Character Recognition (OCR) is the process which optically scans text image and converts it to a machine editable text form. OCR is a real world application of technology where in conversion of printed text to digitized and editable form is performed to make storing and processing of scanned data an easy task. This technology has multiple successful deployments in numerous areas viz. text detection of addresses on envelops and signatures on bank cheques, archiving of data in companies and institutions ancient script retrieval, aid to blinds, etc. Applications of OCR span to include Natural Language Processing, Information retrieval systems and Corpus collection for machine translations, creating digital libraries etc.[1]. The OCR system for Roman scripts like English has attained good level of performance while it still remains an immature area demanding sincere research efforts in context of Devanagari scripts. With over 258 million speakers of languages derived from this script, a detailed exploration of avenues to improve the OCR system for it is in order. Hindi being commonly and popularly used Devanagari script derivative, it has been considered for all experimentations in our work. The OCR system implementation is broadly identified as a constitution of four major stages viz. Pre-processing, Segmentation, Feature extraction and Classification. Pre-processing is done to increase the quality of the image required to allow the steps following it to deliver accurate results. Scanning of text document loading to conversion of paper document into text image is performed prior to Pre-processing. Following preprocessing is the stage of segmentation where in focus is on atomizing the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

data at various level like sentences, words and characters. High degree of accuracy is to be maintained in these two stages for any error in these stages would result in cascade effect and thereby produce a multiplied growth in steps following it which may comprise of feature recognition, extraction and classification.

This work is dedicated to the first two stages owing much to their vitality. No work thus for reports the exploration of Image binarization using Morphological operator for Pre-processing and Active Contour Model (ACM) for word segmentation of Indian scripts, especially Devanagari, as far as authors knowledge extends. Towards this end, we propose a comparative study that aims to bring out the efficiency of the proposed methodology over conventional strategies to pre-processing and segmentation followed for Devanagari scripts. The conventional methods used for this script and other sisterly scripts viz. Gurumukhi script (script of Punjabi language) and languages like Tamil, Hindi, etc. Most of these report the usage of Vertical projection profile (VPP) for word and character segmentation. It is found that both VPP and ACM are equally competent. However, ACM outwits VPP on noisy documents with severe noise contents like complement color patches (speckles), border artifacts etc. A combination of Morphological operator for image binarization, Connected component analysis for border noise removal, Horizontal projection profile for line segmentation and ACM for word and character segmentation has demonstrated significant potential in handling OCR for Devanagari script. The study incorporates robustness evaluation of the fore mentioned by subjecting it to document with varying degree of noise severity and border artifacts. The documents fed to an OCR can be broadly classified into printed and handwritten text. For all purpose of illustration and experimentation, we consider printed text as the source document fed into the OCR system.

The remainder of this paper is organized as follows. Section 2 presents a brief review of some of the popular related works. Section 3 presents the overview of the characteristics of Devanagari script. It discusses about the vowels, consonants, and structure of Devanagari word. A detailed discussion on pre-processing and segmentation methodologies has been reported in Section 4. Section 5 presents an intensive experimentation on effectiveness evaluation of the proposition. Standard measures of effectiveness evaluation have also been included to validate the efficacy. We conclude the paper by presenting conclusion and few important remarks along with the directions of future work on Section 6.

2. RELATED WORKS

Improving efficiency is the major concern in OCR systems. All efforts in OCR technology concentrate on this property. The OCR technology, be it for Roman scripts or Indian, follow the same basic methodology of pre-processing, segmentation, feature detection and extraction, and classification as referred by [2]. In this approach vertical and horizontal projections are used for line, word and character segmentation which obtain a performance result of 93%. The [3] proposes an OCR system comprising of pre-processing by binarization and size normalization by trial and error methods. They perform segmentation using projection profile and report a recognition rate of 87%. Much of the exploration on OCR system's efficiency improvement, in Indian context, has revolved around the exploration of using Otsu's method for preprocessing. The Otsu's thresholding algorithm is the basic thresholding technique used popularly for binarization in most works. The [4] discusses a recursive Otsu thresholding method to get a better binarization result in degraded document. Global thresholding algorithms are usually faster because they use a single threshold based on the global histogram of the gray-value pixels of the image. This method is an improvement to existing techniques to create a novel and effective way to binarize historical documents. The [5] proposes a method for binarization of image using its texture features. They use Otsu thresholding iteratively to produce a threshold values, and texture feature associated with each threshold are retrieved using run length algorithm. They report an improvement of 8.1% over the original Otsu's algorithm. A new method for the binarization of the image other than Otsu's is proposed by [6] in which binarization is done using Morphological operators. Morphological operators are very efficient in that it performs image binarization effectively by taking care of complement color patches in the image. The Morphological operator serves better in case of noisy corpus, especially when border noise is present in abundant. Statistics show that thresholding method fails completely in fore mentioned cases. This is done by using two filters based on two operations dilation and erosion [6, 7]. It gives a much better result than Otsu's binarization method by removing of the background noises much effectively. The [8]

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

discusses active contour models and presents a new algorithm for fast global minimization for Active Contour Model. The basic idea of the Active Contours is to evolve a curve in a given image, and the evolution should stop when the curve meets an object or boundary. Active contour model is an effective model for segmentation [9].

In Indian context, much of the work report the use of horizontal projection profile for line segmentation, vertical projection profile for word, and a combination of both for character segmentation. For instance, [10] discuses a complete OCR for Tamil in which all constituent processes of OCR are described. The thesis reports image binarization being done using morphological operators, skew correction using Hough transform, border noise removal using connected component analysis and Word Segmentation using k-means clustering and character segmentation using connected component analysis. The performance of the proposed method was reported to be validated against an in-home built large scale database taken from Tamil books and other documents including different skew angles and border artifacts. The experimental results show an accuracy of 80% which is significant considering the obscurities involved in the process of Tamil word and character segmentation. A recent work contributed by [11, 12] discuss a line based segmentation using horizontal histogram and word segmentation by vertical projection to detect the boundary of each word. The later performed segmentation on Devanagari and Gurumukhi. They report an accuracy of 98.89 % for character segmentation of Devanagari script. It is seen that most of these works report an accuracy falling between 97% and 100%. However, it does not become explicit in their work as to what kind of document has been used to evaluate the efficacy and robustness of their proposition. Discussion on the degree of noise content in the considered documents has been relatively ignored. Moreover, many of the work fail to report their strategy of preprocessing on documents. The work motivates the exploration on scaling its methodology with some modifications to other scripts like Devanagari, since much of the work on OCR for Devanagari involves the use of conventional approach to OCR for Roman script.

Besides this, as mentioned previously, most of the work do not present detailed discussion on preprocessing steps, noise removal (especially border noise), and the type of document and its noise severity which have been exploited for the efficacy evaluation of their proposition. Moreover, in Indian context, none of the work reported thus far use Active Contour Model (ACM) for word and character segmentation which seems to possess significant potential in handling a script like Devanagari owing much to its segmentation approach. The Vertical Projection Profile has remained a conventional approach to the same. Toward this end, we propose a study highlighting the usefulness of ACM in word and character segmentation in Devanagari script. The study shows that a combination of Morphological operator for image binarization, Connected component analysis for border noise removal, Horizontal projection profile for line segmentation and ACM for word and character segmentation has demonstrated significant potential in handling OCR for Devanagari script. The study incorporates robustness evaluation of the fore mentioned by subjecting it to document with varying degree of noise severity and border artifacts.

3. CHARACTERISTICS FEATURES OF DEVANAGIRI SCRIPT

Devanagari script has 11 vowels and 33 consonants [12]. The same has been illustrated in Figure 1.

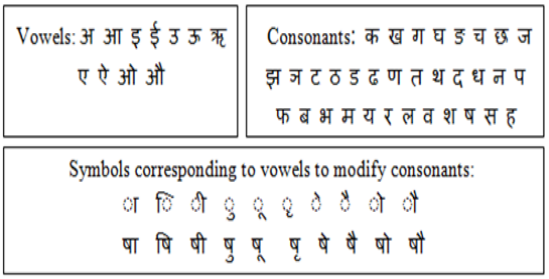


Figure 2: Vowels And Consonants Of Hindi (A Devanagari Script)

A Devanagari word can be divided into three strips viz. core strip, top strip, and bottom strip as illustrated in Figure 2.

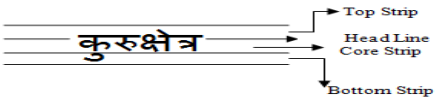


Figure 1: Divisions In The Structure Of Devanagari Word

The core strip and top strip are differentiated by the head line called shirorekha (a Hindi term meaning headline). The lower modifier is attached to the core character which is in the bottom strip.

OCR for Devanagari script becomes even more difficult when compound character and modifier

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

characteristics are combined in 'noisy' situations. A syllable may be formed by composition of a few consonant and vowel sounds. Most of the time, vowels appear as vowel modifiers, i.e., they deform the consonant shape to articulate the composite sound. Some of the combination of conjuncts with characters are depicts in Figure 3.

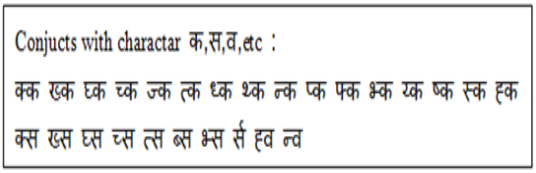


Figure 3: Illustration Of Conjuncts – Character Composition

This creates the need for splitting the syllables into their constituent components which is a nontrivial task. If one considers all the possible characters that can be generated by the composition of consonants and vowels, it may result in a large number of classes. For Hindi, the number of classes then becomes more than 10000. In the case of Devanagari script, most of the modifiers (matras) appears either above the shirorekha or below the character, it is possible to segment out such modifiers and reduce the number of classes to less than 100. [13]

4. PROPOSED METHODS

This section brings out an effective strategy to quality preprocessing and segmentation stage of an OCR system for Devanagari script. The steps involved are explained in Figure 4.

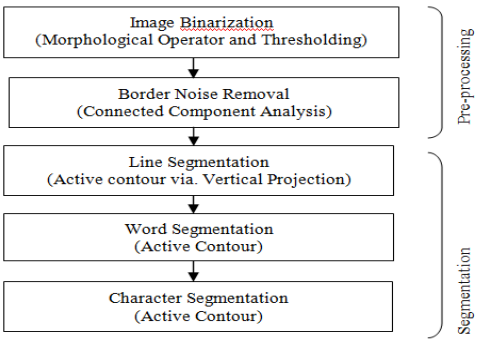


Figure 4: Flowchart Of The Algorithm Under Study

The pre-processing stage handles steps, Image Binarization and Noise Removal. The segmentation stage in OCR performs line, word, and character atomization. Detailed description of the same is presented in subsections below.

4.1 Pre-processing

4.1.1 Image binarization

In this process, a grey scale image is converted to binary image containing only pixel values of ones and zeros. This separates the area of interest from the background information which is essential. The method to binarization is to change the pixel values of area of interest (here greatly characters or text) to higher value i.e, to one and to make all other part zero by fixing a threshold value. The pixels values below this threshold are to be converted to zero and above this threshold to one. The method for binarization applies morphological operator along with thresholding.

4.1.1.1 Morphological operators

Morphology is a term that refers to the description of shape and area of an object. Morphological operators usually process the objects in input image according to its shape. These operators apply a structural element on the objects of the image and produce an output image of the same size. The value of each pixel in the output image of morphological operations is based on the comparison of the corresponding pixels and its neighboring pixels.

The basic operations using morphological operators are dilation and erosion. The former makes the boundary of the image undergo an expansion by addition of pixels to the boundary region and the later results in boundary shrinkage due to the removal of pixels from the boundary region. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image.

4.1.1.1.1 Dilation

The value of the output pixel is *maximum* value of all pixels in input pixel's neighborhood. As regards to binary image, if any of the pixels is set to the value 1, the output pixel is set to 1. The dilation of a gray scale image is illustrated in Figure 5.

The structuring element defines the neighborhood of the pixel of interest, which is circled. The morphological dilation function sets the value of the output pixel 1 because one of the elements in the neighborhood defined by the structuring element is one.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

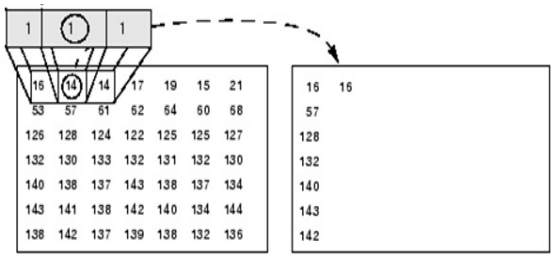


Figure 5: Dilation Of A Gray Scale Image

Erosion

The value of the output pixel is *minimum* of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set 0. The erosion of a gray scale image is illustrated in Figure 6.

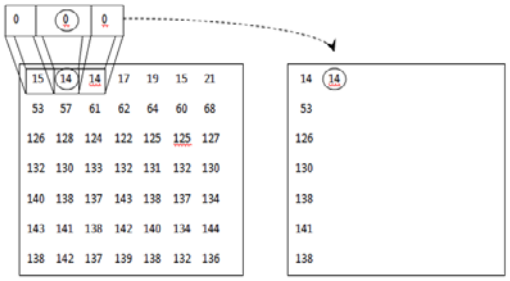


Figure 6: Erosion Of A Gray Scale Image

There are two operation which is performed on combing erosion and dilation operation viz.
"Morphological Opening" of an image which is a successive operation of erosion followed by a dilation, using the same structuring element for both operations and "Morphological Closing" operation is a successive operation of dilation and then erosion using a same structuring element. The former eliminates thin protrusion and removes small object from an image while preserving the original image and later tends to smooth sections of contour and fuses narrow breaks and eliminates small holes.

4.1.1.2 Algorithm

Binarization is done on grayscale images. On grey scale image, the morphological closing operation is done after this the output of closing operation got subtracted from the original grey scale image. Then a global thresholding is done with a threshold of value 50 to get a binary output image. The algorithm is described using a flowchart as shown in Figure 7.

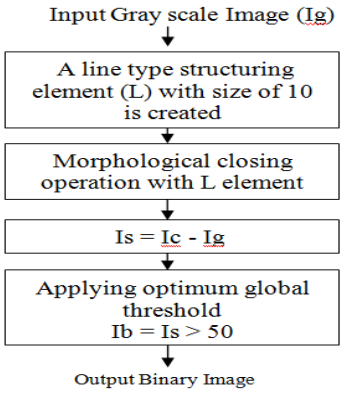
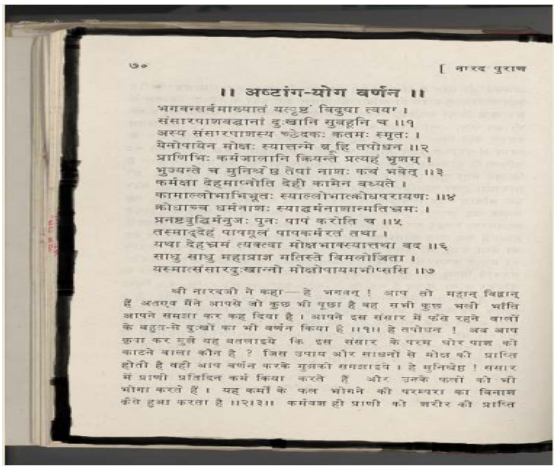
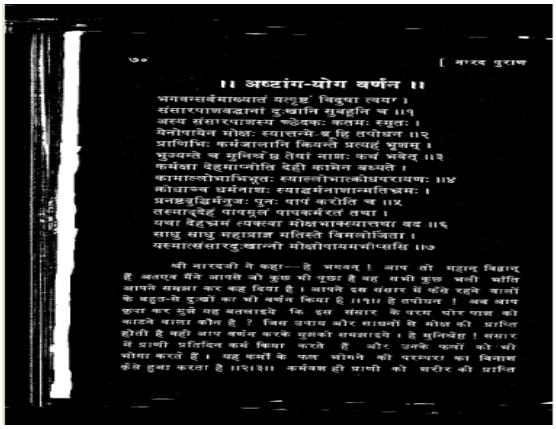


Figure 7: Algorithm For Binarization

An original image and its binarized form is shown in Figure 8.



Orginal Image



Binary Image Output Using Morphological Operators
Figure 8: Binary image output

The method gives text quality similar to Otsu's method. However, if any noise is present usually

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

page border or border noise, this will result in a less noise output.

4.1.2 Noise removal

Second step in preprocessing is noise removal were the noises are removed; here the noise under consideration is border artifacts. Since the input is a scanned image and chances are high that the image is a part of large document, borders will be formed for these scanned text images. This will take as a valid data during binarization which is actually a noise which has to be removed.

4.1.2.1 Border artifacts and page borders

Border artifacts can be dark lines along the page borde r or speckles that result from the binarization process. Particularly when we scan thick books, we see the dark impression around the borders. The page borders are also found as dark thick lines which will always taken into consideration as true object which are actually noise. This scenario is depicted in Figure 8. The presence of such vertical and horizontal lines decreases the accuracy of OCR considerably and results in improper segmentation. Connected component algorithm is proposed for the elimination of these border noises. The algorithm is explained in the following section.

4.1.2.2 Connected component analysis

Also called as blob detection or region extraction, CC is commonly used analysis in image segmentation algorithms. It scans an image and groups pixels based on its connectivity. Generally CC analysis is done over binary images. The output of CC analysis gives a label matrix having same dimension as of the original image.

The example of a labeled matrix is shown in Figure On obtaining label matrix, it is easy to extract each of the components to perform separate processing. The result of CC analysis for Hindi character ' क ' is shown in Figure 10.

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	3	3	0
1	1	1	0	0	0	0	0

Figure 9: Labeled Matrix

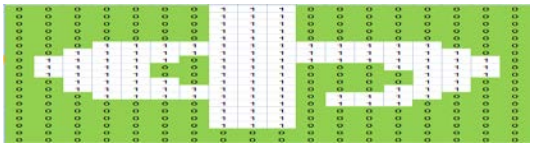


Figure 10: CC Analysis

In below CC analysis for border noise removal is described..

4.1.2.3 Algorithm

Morphological closing operation had done in the document image with a structuring element of rectangle. The size of the rectangle is taken very small. For vertical borders (left and right), it is "[3 1]" and for horizontal borders (top and bottom) it is "[1 3]". It helps in smoothing the document borders. For each border, we apply CC analysis and then extract out the largest component present. We take features (height) of the largest CC. This is validated against a given threshold and a decision is taken whether it's a border noise or not. If that CC is a border noise then all the pixels are made zero. The algorithm is described in detail in Figure 11

.

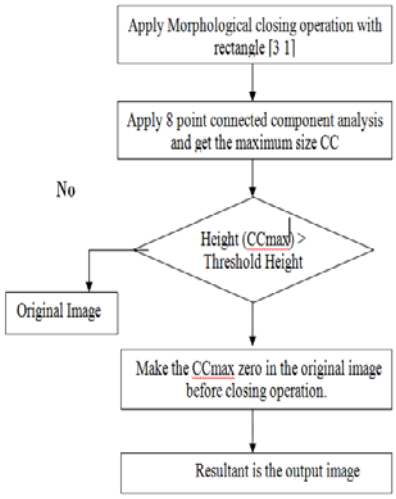
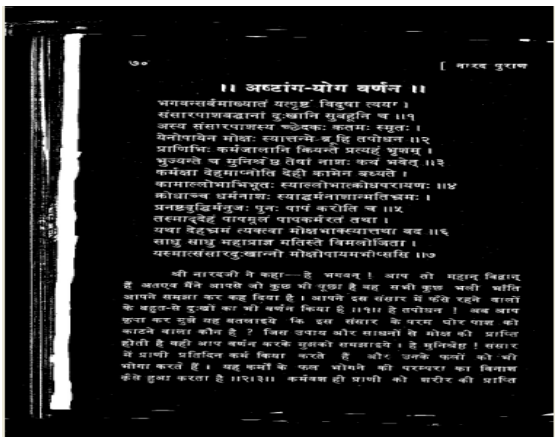


Figure 11: Algorithm Border Removal

The images before and after border removal are shown in Figure 12.



Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 12: Image Before And After Border Removal

An effective pre-processing is followed up by the process of segmentation. A detailed approach to various levels of segmentation is presented the succeeding subsection.

4.2 Segmentation

4.2.1 Line segmentation

In line segmentation, the focus is on separating individual lines in a script document. The method involves construction of a horizontal histogram of the image like the one shown in Figure13.

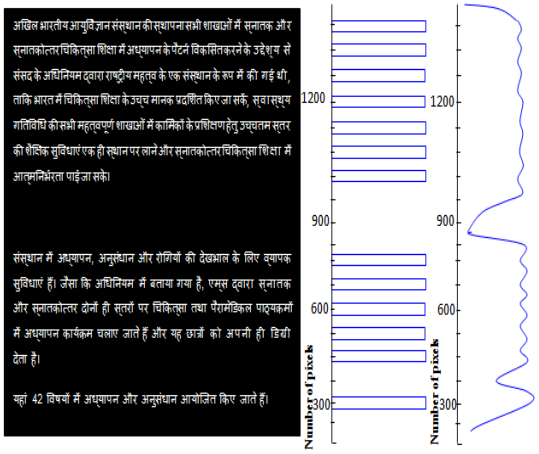


Figure 13: Horizontal Projection Profile

Based on the peak of this horizontal Histogram individual lines in the document image are separated. The details of Horizontal profile projection are explained below .

4.2.1.1 Horizontal projection

The HPP is calculated by summing the white pixels in each row of the image. The HPP graph contains peaks and valleys symbolizing the text lines and line gaps respectively.

4.2.1.2 Algorithm

Construct the Horizontal Histogram for the image. Using the Histogram, find the points from which the line starts and ends. For a line of text, upper line is drawn at a point where we start finding white pixels and lower line is drawn where we start finding absence of white pixels. And the process continues for next line and so on. The horizontal projection profile of binary document image is represented by Figure 13.

4.2.2 Word segmentation

Word segmentation aims to resolve individual word in a script document. This is done based on the boundary of each word. The boundary of each word is identified and word separation is done according to it. Active Contour is the proposed alternative. Vertical profile projection is also discussed and performed in order to do a comparative study on deteriorated Images.

4.2.2.1 Active contours

An active contour is an energy minimizing spline that detects specified features within an image. It is a flexible curve (or surface) which can be dynamically adapted to required edges or objects in the image (it can be used to automatic objects segmentation). It consists of a set of control points connected by straight lines. The active contour is defined by the number of control points as well as sequence of each other. Fitting active contours to shapes in images is an interactive process.

The Active contour model helps to trace every closed shape with in an image. By using this property of the model every boundary of a word with in a line can be traced, since each word is a closed shape within a line. Using this each word can be separated as a different image. The binarized image and corresponding word segmented form is shown in Figure 14.

Figure 14: Active Contour Tracing On A Line To Get Word Segments

4.2.2.2 Vertical projection

The VPP is calculated by summing the white pixels in each column of the image. The VPP graph contains peaks and valleys symbolizing the words and word gaps respectively in a line. For VPP, image of each line is taken and is scanned vertically to get the non-zero pixel values. The columns where non-zero values are absent are considered as the gaps between the words and the segmentation is performed at these points.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2.3 Character segmentation

In character segmentation individual character in a script document image is separated based on the extraction of the word. Yet again, Active Contour is the proposed alternative for character segmentation. By using the same procedure as for word, each character with in a word is also separated after shirorekha removal using horizontal projection as shown in Figure 15.

Figure 15: Word Before And After Shirorekha Removal

The character tracing after shirorekha removal is shown in Figure 16.

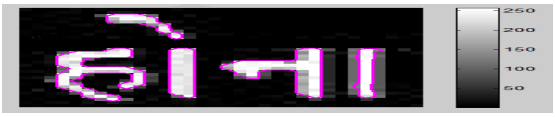
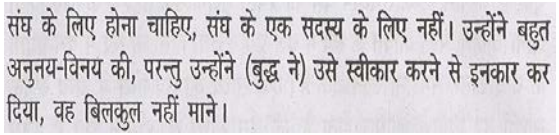


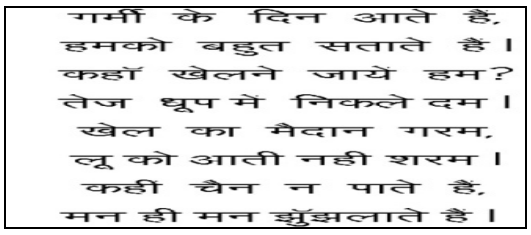
Figure 16: Active Counter Tracing For A Word To Get Character Segments

4. EXPERIMENTATION AND RESULT STATISTICS ANALYSIS

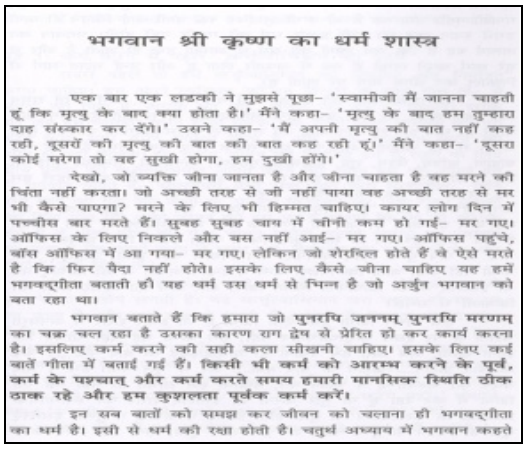
A detailed account of the experimentation on effectiveness evaluation of the segmentation algorithm has been described in this section. The focus of the experimentation is to demonstrate the potential of Active contour based segmentation for Devanagari script. Hindi has been chosen as the candidate for the experimentation. In order to prove the efficiency and robustness of the proposed alternative algorithms, we have subjected it to diverse documents with varying degree of noise severity and other intense border artifacts. These documents have been shown in Figure 17.


संघ के लिए होना चाहिए, संघ के एक सदस्य के लिए नहीं। उन्होंने बहुत अनुनय-विनय की, परन्तु उन्होंने (बुद्ध ने) उसे स्वीकार करने से इनकार कर दिया, वह बिलकुल नहीं माने।

Doc-3


गर्मी के दिन आते हैं,
हमको बहुत सताते हैं।
कहाँ खेलने जायें हम?
तेज धूप में निकले दम।
खेल का मैदान गरम,
लू को आती नहीं शरम।
कहीं चैन न पाते हैं,
मन ही मन झुँझलाते हैं।

Doc-1


भगवान श्री कृष्ण का धर्म शास्त्र

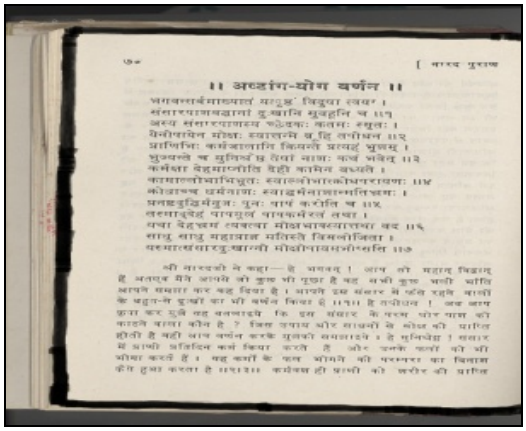
एक बार एक लडकी ने मुझसे पूछा– 'स्वामीजी मैं जानना चाहती हूं कि मृत्यु के बाद क्या होता है।' मैंने कहा– 'मृत्यु के बाद हम तुम्हारा दाह संस्कार कर देंगे।' उसने कहा– 'मैं अपनी मृत्यु की बात नहीं कह रही, दूसरों की मृत्यु की बात की बात कह रही हूं।' मैंने कहा– 'दूसरा कोई मरेगा तो वह सुखी होगा, हम दुखी होंगे।'

देखो, जो व्यक्ति जीना जानता है और जीना चाहता है वह मरने की चिंता नहीं करता। जो अच्छी तरह से जी नहीं पाया वह अच्छी तरह से मर भी कैसे पाएगा? मरने के लिए भी हिम्मत चाहिए। कायर लोग दिन में पचासों बार मरते हैं। सुबह सुबह चाय में चीनी कम हो गई– मर गए। ऑफिस के लिए निकले और बस नहीं आई– मर गए। ऑफिस पहुंचे, बॉस ऑफिस में आ गया– मर गए। लेकिन जो शेरदिल होते हैं वे ऐसे मरते है कि फिर पैदा नहीं होते। इसके लिए कैसे जीना चाहिए यह हमें भगवद्गीता बताती है यह धर्म उस धर्म से भिन्न है जो अर्जुन भगवान को बता रहा था।

भगवान बताते हैं कि हमारा जो पुनरपि जननम् पुनरपि मरणम् का चक्र चल रहा है उसका कारण राग द्वेष से प्रेरित हो कर कार्य करना है। इसलिए कर्म करने की सही कला सीखनी चाहिए। इसके लिए कई बातें गीता में बताई गई हैं। किसी भी कर्म को आरम्भ करने के पूर्व, कर्म के पश्चात् और कर्म करते समय हमारी मानसिक स्थिति ठीक ठाक रहे और हम कुशलता पूर्वक कर्म करें।

इन सब बातों को समझ कर जीवन को चलाना ही भगवद्गीता का धर्म है। इसी से धर्म की रक्षा होती है। चतुर्थ अध्याय में भगवान कहते

Doc-2


॥ अरुंधति-योग वर्णन ॥

Doc-4

Figure 17: Documents used for Robustness Evaluation.

These documents have been described as follows:

Doc-1: Noiseless document (0% noise).

Doc-2: Partially visible contents of next page (Thin page artifact).

Doc-3: Multiple complement color patches/speckles at back ground.

Doc-4: Scattered border noise with partially visible contents of next page.

Standard measures of effectiveness evaluation have been used for efficiency evaluation viz. Recall and Precision. These have been defined below:

$$\text{Recall} = \frac{\text{Total no. of images correctly retrieved}}{\text{Total no. of correct images}} \quad (1)$$

$$\text{Precision} = \frac{\text{Total no. of images correctly retrieved}}{\text{Total no. of retrieved images}} \quad (2)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 1: Result Of Line Segmentation For Devanagari Script Using Proposed Alternatives.

Document Images	No. of lines present	Output of line segmentation	No. of lines correctly segmented	Recall (%)	Precision (%)
Doc-1	8	8	8	100	100
Doc-2	23	23	23	100	100
Doc-3	3	3	3	100	100
Doc-4	26	26	26	100	100

TABLE 2: Result Of Word Segmentation For Devanagari Script Using Proposed Alternatives.

Document Images		Doc-1	Doc-2	Doc-3	Doc-4
No. of words present		42	314	40	233
Output to word segmentation (Active Contour method)		42	336	43	293
No. of words correctly segmented (Active Contour method)		42	311	40	202
Output to word segmentation (Vertical projection profile method)		42	278	30	181
No. of words correctly segmented (Vertical projection profile method)		37	242	23	146
Recall (%)	Active Contour	100	99.04	100	86.69
	Vertical Projection	88.09	77.07	57.5	62.66
Precision (%)	Active Contour	100	92.55	93.02	68.94
	Vertical Projection	88.09	87.05	76.66	80.66

The result statistics obtained on subjecting the various documents to the proposed alternatives is discussed in the remainder of this section. Table 1 shows the result obtained upon using the HPP for line segmentation in various documents. Like in any other script, HPP has delivered cent percent recall and precision on effectively pre-processed Hindi (Devanagari script) documents.

The obtained statistics out of the experimentation conducted for word segmentation on the four types of *documents* is shown in Table 2. The word segmentation has been carried out using ACM and VPP. The recall and precision values obtained for Doc-1, 2 and 3 using ACM exceeds the corresponding figures obtained using VPP by a huge margin, as evident from Table 2, especially in Doc-1 which is noise free.

Figure 18: Comma Almost Attached To The Word Noise In Between

An instance to justify the reason for VPP to perform poorly is illustrated in Figure18 which shows the result for word segmentation using VPP. It is evident that this conventional methodology completely fails to separate comma (,) from the main word "है". Similarly, other characters like colon, full stop, etc. are also not separated using VPP due to its global threshold condition which greatly reduces the accuracy of VPP. The problem is effectively overcome using ACM for it identifies closed loops as characters. Thus, "है" and "," will be rightly separated using ACM. Further, an appreciably good and significantly better figure of recall and precision is obtained using ACM for Doc-2 and Doc-3. An important instance which brings down the accuracy of VPP is illustrated in Figure 19 which shows presence of pixel noise as complement color patches between two words. VPP regards the two words in Figure19 as a single unit due to its thresholding conditions which require a minimum distance between two words to be identified as separate entities.

Journal of Theoretical and Applied Information Technology
20th June 2013. Vol. 52 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

TABLE 3: Result Of Character Segmentation For Devanagari Script Using Proposed Alternatives.

Document Images	No. of characters present	Output of character segmentation	No. of characters correctly segmented	Recall(%)	Precision(%)
Doc-1	101	103	100	99.00	97.08
Doc-2	781	1005	780	99.87	77.61
Doc-3	99	102	99	100	97.05
Doc-4	975	1181	967	99.19	81.87

TABLE 4: Result Of Top Character Segmentation For Devanagari Script Using Proposed Alternatives.

Document Images	No. of Top characters present	Output of Top character segmentation	No. of Top characters correctly segmented	Recall (%)	Precision (%)
Doc-1	27	27	27	100	100
Doc-2	223	223	223	100	100
Doc-3	29	29	29	100	100
Doc-4	173	173	171	98.84	98.84

Figure 19: Words with white pixel

The minimum distance is not encountered due to presence of noise pixels. However, ACM identifies the two words correctly apart from identifying each pixel as an individual unit. This partially degrades its precision, but its recall remains high. In Doc-4, with most severe noise content, the precision using ACM goes low on account of factors like speckles being identified as words. However, recall remains high since all valid segments get captured by ACM. It can be seen that ACM identifies 202 word segments while VPP identifies only 146, out of actual 233 words present in the document. This makes VPP less popular and inefficient in case of noisy document. The reason for large segments obtained using ACM is attributed to its methodology, i.e., identification of closed curves. However, this property of ACM may fail to correctly segment some rare words in Hindi containing "visarg" i.e., the symbol ':' at the end of the word. An example illustrating such a condition is shown in Figure 20.

Figure 20: Illustrating Fail Factor Of Active Contour Model

In this case the identified words will be nine while it is actually three. This is due to the fact that ACM recognizes closed loops as words. Since each "visarg" is identified as a separate word, it contributes to an erroneous segmentation and the accuracy of ACM is hampered depending on their frequency of occurrence in the scanned documents. On contrary, VPP delivers right results in this scenario on account of its global thresholding condition that does not allow the separation of a "visarg" due to its close proximity to the word it is attached to. With such a condition being rare in Hindi, one may ignore the fore mentioned and conclude that ACM is a better choice for word segmentation.

Table 3 discusses the character segmentation statistics obtained upon using ACM for character segmentation. The character segmentation using ACM gives an accuracy of 100%.

The statistics obtained for character and top character segmentation using ACM far exceed their counterparts obtained using VPP. The statistics and discussion on the later have not been included due to space constraints. However, it is perfectly intuitive that VPP should perform inferior to ACM for same has been the scenario in word segmentation. It was seen that VPP failed to perform correct inter-word segmentation on account of global thresholding conditions. The same phenomenon was observed in our experimentation for intra-word segmentation (character segmentation) using VPP.

A dip in the precision is observed in document Doc-2 and Doc-4 in Table 3. This occurred on account of presence of background noise and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

complement color pixel/speckle noise. Since each of these junk characters are also falsely recognized as valid segments, the precision goes low. It is worthwhile to note that recall remains consistent for correct segments are rightly identified by ACM. Another problem which hampered system precision is character overlapping in original source documents. The result of character segmentation of the word "निकले" present in Doc-1 which illustrates the above discussed problem is shown in Figure 21.

Figure 21: Illustrating Fail Factor Of Acm (And Vpp) Due To Character Overlap

The word "निकले" contains three valid segments i.e., "नि", "क" and "ले". The original hand written document had "क" and "ले" partially overlapped. This enables ACM identify "नि" and "कल" alone as character segments from scanned documents. This is also the reason attributed to degraded recall for Doc-1 on Table 3. These are purely source document artifacts and cannot be handled by ACM or any other segmentation algorithm including VPP. Table 4 presents the statistics obtained for top character segmentation using ACM.

The results obtained using ACM algorithm has demonstrated significant potential in handling effective top-character segmentation on all types of documents, as evident from the tabulation. The result for Doc-4 falls short of cent percent due to the reason described using Figure 21.

The upper portion of the shirorekha of the word "हैं" in Doc-4 is shown in Figure 22. Since the original hand written document had "ै" and "ं" overlapped, it could not be separated using ACM.

From the above discussion, it is evident that proposed combination of methods for preprocessing and segmentation provides improved results for Devanagari script segmentation over other conventional approaches.

6. CONCLUSION

This paper focuses on subjecting exposure to alternate existing methodologies to pre-processing and segmentation of Devanagari script (printed text). An intensive study and quantified justification has been reported to testify the efficiency of the proposed alternatives. It has been shown that a combination of Morphological operator for image binarization, Connected component analysis for border noise removal, Horizontal projection profile for line segmentation and Active contour model for word and character segmentation works best for Devanagari script. The robustness of the proposed alternative has been demonstrated by validating its efficiency on multiple documents with varying degree of noise severity and border artifacts. The experimental results confirm our proposition to be a superior technique over other conventional methodologies to pre-processing and segmentation in OCR system implementation. The efficiency of ACM on Devanagari script owes much to its approach to segmentation. ACM functions superior to VPP since it considers closed loops for segmentation. It effectively identifies all commas and full stops in the text which are not separated in VPP on account of thresholding conditions. Also, it has been found that both VPP and ACM are equally competent on noiseless documents. However, ACM has delivered consistent performance on noisy documents containing foul characters like complement color patches (white pixels or speckles), border artifacts, etc. Owing to inadequate preprocessing a dip in system precision was observed on few noisy documents. However, recall obtained using ACM remained 100% on all types of documents for all valid segments were identified correctly. Consequently, it shall be interesting to determine if modifications could be made on Morphological operator to enhance its cleaning effect on noisy documents. Also, it shall be a worthwhile effort to bring out modifications on ACM algorithm to effectively handle segmentation process in an instance of "visarg" or other typical special characters along with eliminating its handicapped-ness on overlapping characters intended to increase the precision on scripts with such artifacts. Our future work shall focus this aspect along with exploring avenues that should further improve the parameters which govern the accuracy of the OCR system.

REFRENCES:

[1] Krishnan P, "Tamil Optical Character Recognition",*Centre for Excellence in Computational Engineering and Networking*, The University of Texas at El Paso, 2011.

[2] Bansal V, Sinha RMK, "A complete OCR for printed Hindi text in Devanagari script", *Sixth International Conference on Document Analysis and Recognition*, 2011, pp. 800-804.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] Desai A, Malik L, Welekar R, "A New Methodology for Devanagari Character Recognition", *International Journal of IT*, Vol. 1, 2011, pp. 626-632.

[4] Nina O, Morse B, Barrett W, "A recursive Otsu thresholding method for scanned document binarization", *IEEE Workshop on Applications of Computer Vision (WACV)*, 2011, pp. 307-314.

[5] Liu Y, Srihari SN, "A recursive Otsu thresholding method for scanned document binarization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 191,1997, pp. 540-544.

[6] Brodic D, Milivojevic DR, Tasic V, "Preprocessing of Binary Document Images By Morphological Operators", I*n MIPRO 2011 Proceedings of the 34th International Convention* ,2011, pp. 883-887.

[7] Shafait F, Breuel TM, "A simple and effective approach for border noise removal from document images", *IEEE 13th International Multitopic Conference INMIC*, 2009, pp. 1-5.

[8] Bresson X, ". A Short Guide on a Fast Global Minimization Algorithm for Active Contour Models", *IEEE Transactions on Pattern Analysis*, 2009.

[9] Zhang XL, Shi ZG, "A new algorithm for text segmentation based on stroke filter", *In Control and Decision Conference (CCDC)*, 2010, pp. 4347-4350.

[10] Ramanathan R, Ponmathavan S, Valliappan N, Nair TL, Soman KP, "Optical Character Recognition for English and Tamil Using Support Vector Machines", In International Conference on Advances in Computing, Control, & Telecommunication Technologies, 2009, pp. 610-612.

[11] MK, Patnaik T, Tiwari S, Singh SK, "Script Segmentation of Printed Devnagari and Bangla Languages Document Images OCR", *International Journal Of Computer Science and Technology*, 2011.

[12] Kumar V, Kumar P. Sengar," Segmentation of Printed Text in Devanagari Script and Gurmukhi Script", *International Journal Of Computer Applications, Vol. 3*, 2010.

[13] Jawahar CV, Pavan Kumar MNSSK, Ravi Kiran SS," A bilingual OCR for Hindi-Telugu documents and its applications", *In Seventh International Conference on Document Analysis and Recognition,2008,pp.* 408- 412.