ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

Assessment of Optical Character Recognition Techniques for Hindi Language

M Kalidas[1], Madhu Bajaj[2]

Assistant Professor, Department of Master of Computer Applications, CBIT, Gandipet, Hyderabad, India[1]

R&D Engineer Software 1, Broadcom Inc., Hyderabad, India[2]

ABSTRACT: Optical Character Recognition(OCR) is one of the actively researched areas in both industry and academia because of its potential applications. OCR is a widely used technology to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of images containing written text (typed, handwritten or printed) into machine-readable encoded text data. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, text-to-speech and text mining. OCR engines are used to read typed (machine printed) characters. At present, there are several OCR engines available for usage. One such OCR engine is Tesseract. Tesseract is an optical character recognition engine that can be used with various operating systems. It can recognize over 100 different languages. It does various image processing operations internally before carrying out the actual OCR. Tessaract performs a reasonably good OCR, but there are certain cases where it is not good enough, which can result in a significant reduction in accuracy. The present work involves using different predefined functions and features of MATLAB and various techniques available in MATLAB for improving the efficiency and performance of Optical Character Recognition (OCR) for the Hindi language. The results of this work are subjected to comparison with an existing OCR engine.

KEYWORDS: MCR, Tesseract, MSER

I. INTRODUCTION

Character recognition can be broadly categorized into online and offline methods. In online text recognition, characters and words are recognized real-time as soon as they are written, and therefore, contain temporal information. Online systems obtain the position of the pen as a function of time directly from the interface. This is usually done through pen-based interfaces where the writer writes with a special pen on an electronic tablet.

Off-line handwriting recognition refers to the process of recognizing words that have been scanned from a surface (such as a sheet of paper) and are stored digitally in greyscale format. After being stored, it is conventional to perform further processing to allow superior recognition. The major difference between Online and Offline Character Recognition is that Online Character Recognition has real-time contextual information but offline data does not [2].

An Offline text recognition system processes a static representation of a document. Offline text recognition is divided into two subcategories of typed and handwritten documents. In both subcategories, an image of the document obtained from a scanner or camera is processed. Obviously, due to the variety of handwriting styles and non-standard nature of handwriting, the problem of offline handwriting recognition is the most challenging problem in OCR and it usually requires language-specific methods. On the other hand, OCR of typed documents are very much in demand for practical applications such as historical document analysis, official letter and document processing, and vehicle plate recognition.

The offline character recognition can be further grouped into two types: Magnetic Character Recognition (MCR) and Optical Character Recognition (OCR)

In MCR, the characters are printed with magnetic ink. The reading device can recognize the characters according to the unique magnetic field of each character. MCR is mostly used in banks for check authentication.OCR refers to the

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

branch of computer science that involves reading text from paper and images and translating the images into a form that the computer can manipulate (for example, into ASCII codes).

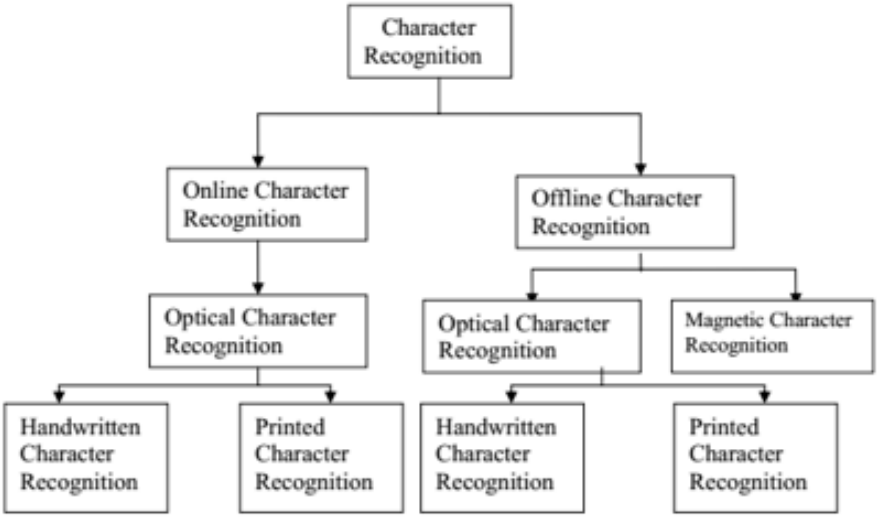

Figure 1: Classification of Character Recognition

II. OPTICAL CHARACTER RECOGNITION

Optical character recognition includes the mechanical and electrical conversion of scanned images of handwritten, or typewritten text into machine text. It is a common method of digitizing printed texts so that they can be electronically searched, stored more compactly, displayed on-line, and used in machine processes such as machine translation, text to speech and text mining[1]. In recent years, OCR (Optical Character Recognition) technology has been applied throughout the entire spectrum of industries, revolutionizing the document management process. OCR has enabled scanned documents to become more than just image files, turning into fully searchable documents with text content that is recognized by computers. With the help of OCR, people no longer need to manually retype important documents when entering them into electronic databases. Instead, OCR extracts relevant information and enters it automatically. The result is accurate, efficient information processing in less time. [2].

It is the efficiency of preprocessing techniques and character recognition algorithms applied prior to giving the image to an OCR engine that increases the performance and accuracy of OCR results. In simpler words, higher the accuracy of input image achieved using preprocessing and other recognition methods, higher is the performance of OCR on that specific image. Traditionally, many preprocessing techniques and algorithms were not developed by then and hence were not available for use which resulted in somewhat less accuracy of OCR results [3].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

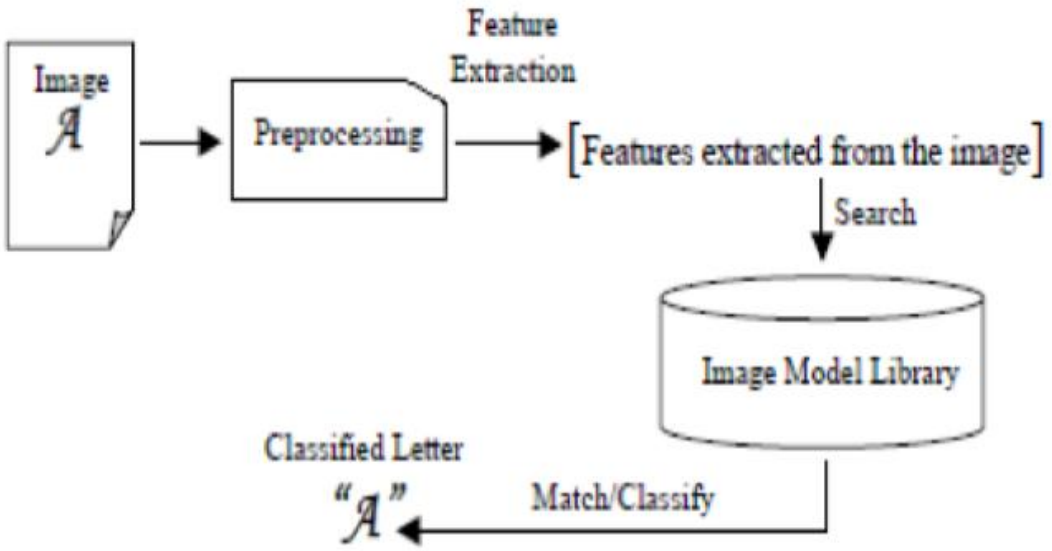

Figure 2: General phases of OCR

The proposed work presents how the performance and efficiency of OCR can be increased with the use of currently available preprocessing techniques for the accurate recognition of Hindi language in comparison to an existing OCR engine Tesseract.

Although Hindi is the national language of India, the majority of the population in the country is not well aware of the language. Majority of the people in South India find it difficult to communicate and interact in Hindi, as they are not familiar with the language. Taking into consideration the importance attributed to the Hindi language in our country, it is necessary and desirable to have an accurate and readily accessible tool that helps people recognize and understand the language easily and efficiently. Although OCR engines are available for character recognition and provide support for various languages, they fail in performing OCR for Indic scripts like Hindi accurately. Therefore, we are in need of improving the accuracy and performance of Hindi OCR in comparison to the existing OCR engines.

III. OCR ENGINES

The text recognition from an image is still a challenging task due to the large variety of text appearing in these images. Text in images can have different variations of the size, font style, distortion; it can have different contrast due to different lighting conditions. As the accuracy of text regions of the image increases with increasing efficient preprocessing methods and algorithms, the accuracy of OCR for that image also increases. But, many existing OCR engines like Tesseract do not incorporate the use of these enhanced techniques and therefore, produce less accurate OCR results for images.[4] This problem is even more severe for images containing non-English text (in this case, Hindi) because the typical OCR engines are developed for recognizing English characters only and produce highly accurate results for OCR for English language only. For recognizing other language characters, the existing OCR engines must be trained accordingly with the respective language packages and datasets, which do not provide accurate results when compared to English. Even after training the existing OCR engines, they fail to recognize Indic scripts and Devanagari scripts like Hindi, Sanskrit, Bangla, etc. to the utmost accuracy because the OCR engines cannot identify many features of Devanagiri script characters like “maatras”, half-characters, general vowels and consonants, numerals, extensions, punctuation marks, and special symbols.[5]

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

IV. METHODOLOGY

The proposed work uses different preprocessing techniques like Grayscaling, Thresholding and Binarization and character recognition algorithms like MSER (Maximally Stable Extremal Regions) for achieving the highest possible accuracy of OCR for Hindi language characters and text, when compared to existing OCR engines which do not make use of these improved preprocessing methods, and hence produce less accurate OCR results for Hindi language characters. MATLAB is used for implementation because of the available inbuilt OCR language packages and functions and predefined methods for grayscaling, binarizing and MSER detection. The results of the work are then compared with that of Tesseract. In the first step, an image containing Hindi text is given as input to both Tesseract and the proposed algorithm in MATLAB. In the second step, the image undergoes various enhanced preprocessing stages before it is given as input to OCR. In the third step, the results from both MATLAB and Tesseract are obtained and are compared with each other.[6]

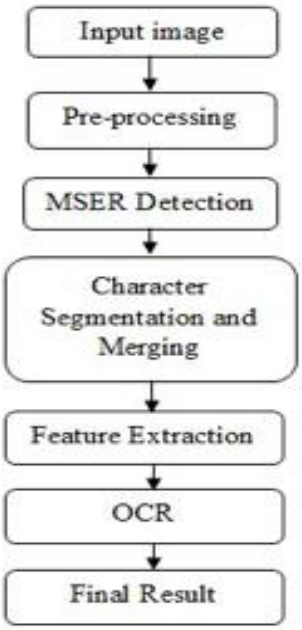

Fig 3: System Design for Proposed OCR System

Proposed work uses Maximally Stable Extremal Regions (MSER) algorithm for detecting text regions from the image, segregating text regions from the non-text regions of the image and then removing non-text regions of the image using:

- Basic Geometric Properties
- Stroke Width Variation

After removing the non-text regions, only the text regions of the image are given as input to OCR. This helps in identifying text in the image correctly, thereby increasing the accuracy of OCR.[6][7]

Text Detection Algorithm:

MSER Algorithm:

Maximally Stable Extremal Regions (MSER) is a feature detector. The MSER algorithm extracts from an image I a number of co-variant regions, called MSERs. An MSER is a stable connected component of some level sets of the image I. Optionally, elliptical frames are attached to the MSERs by fitting ellipses to the regions. MSER is a method for blob detection in images.

MSER is based on the idea of taking regions that stay nearly the same through a wide range of thresholds.[6]

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

·All the pixels below a given threshold are white and all those above or equal are black.
· If we are shown a sequence of thresholded images I with frame **'t'** corresponding to threshold **'t'**, we would see first a black image, then white spots corresponding to local intensity minima will appear then grow larger.
·These white spots will eventually merge until the whole image is white.
The set of all connected components in the sequence is the set of all extremal regions.
The word extremal refers to the property that all pixels inside the MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all the pixels on its outer boundary.
This operation can be performed by first sorting all pixels by gray value and then incrementally adding pixels to each connected component as the threshold is changed.[8]
The area is monitored. Regions such that their variation wrt the threshold is minimal are defined maximally stable:
Considering a sequence of thresholded images with increasing thresholds sweeping from black to white we pass from a black image to images where white blobs appear and grow larger by merging, up to the final image.
Over a large range of thresholds, the local binarization is stable and shows some invariance to the affine transformation of image intensities and scaling.[9]

MSER Processing:
MSER extraction implements the following steps:

1. Sweep threshold of intensity from black to white, performing simple luminance thresholding of the image.
2. Extract connected components. ("Extremal Regions")
3. Find a threshold when an extremal region is **"**Maximally Stable**"**, i.e. local minimum of the relative growth of its square. Due to the discrete nature of the image, the region below / above may be coincident with the actual region, in which case the region is still deemed maximal.
4. Approximate a region with an ellipse (this step is optional)
5. Keep those regions descriptors as features.

However, even if an extremal region is maximally stable, it might be rejected if:
· It is too big (there is a parameter MaxArea)
· It is too small (there is a parameter MinArea)
· It is too unstable (there is a parameter MaxVariation)
· It is too similar to its parent MSER

Margin = the number of thresholds for which the region is stable [10]

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

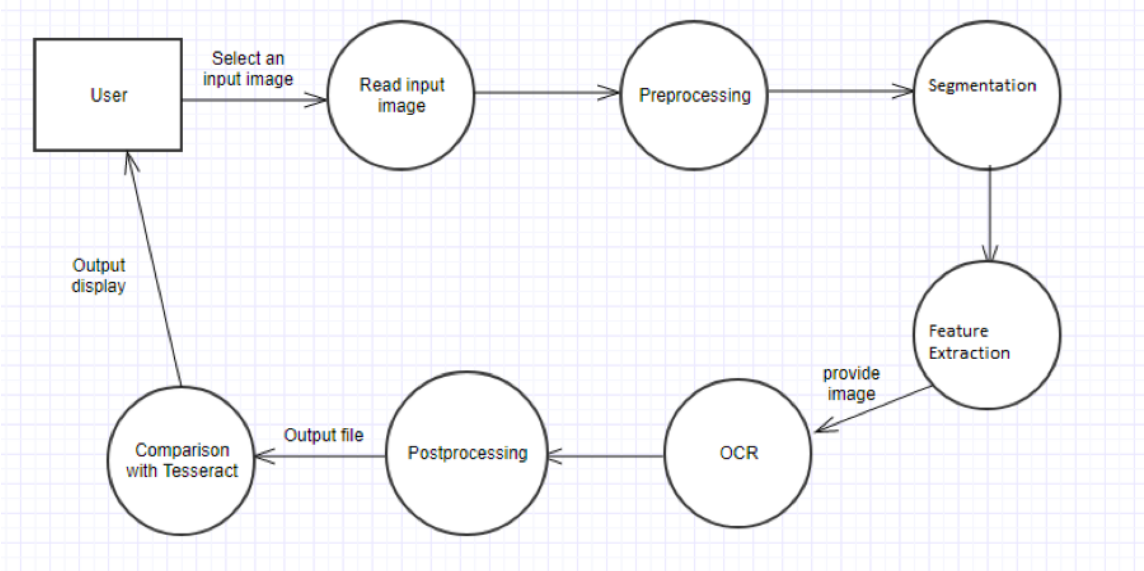

Figure 4: Sequence of steps for OCR and Comparison

V. IMPLEMENTATION

The work involves using different predefined functions and features of MATLAB and various techniques available in MATLAB for improving the efficiency and performance of Optical Character Recognition (OCR) for the Hindi language. The results of this work are subjected to comparison with the existing OCR engine Tesseract. This work comprises of various phases:

·Input image selection
·Conversion of input RGB image to Grayscale image
·Detecting text regions using MSER
·Removing non-text regions
    Basic geometric properties
    Stroke Width variation
    Thresholding
·Merge text regions for final detection region
    Bounding boxes
·Feature Extraction
·OCR and Final results
·Performance Measurement

At first, an image from the current folder is to be selected by the user. This image will be treated as the input image. The input image is then converted into a grayscale format and displayed to the user. This grayscaled image will then be considered for the further operations.

The input image may consist of various text and non-text regions like other images, graphics, hyperlinks, etc. Here, the highest priority is to be given to the text regions of the image as they are to be given as input to OCR in MATLAB. After the input image is selected, the text regions of the image must be segregated from the non-text regions of the image. This is done with the help of Maximally Stable Extremal Regions (MSER).[11]

The next step is to remove the non-text regions from the input image based on basic geometric properties. Although the MSER algorithm picks out most of the text, it also detects many other stable regions in the image that are not text. A

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

rule-based approach is used to remove non-text regions. There are several geometric properties that are good for discriminating between text and non-text regions.[12]

The next step is to remove non-text regions based on Stroke Width Variation. Another common metric used to discriminate between text and non-text is stroke width. Stroke width is a measure of the width of the curves and lines that make up a character. Text regions tend to have little stroke width variation, whereas non-text regions tend to have larger variations.

The following step is to merge text regions for Final Detection Result. At this point, all the detection results are composed of individual text characters. To use these results for recognition tasks, such as OCR, the individual text characters must be merged into words or text lines. This enables recognition of the actual words in an image, which carries more meaningful information than just the individual characters.[13]

After all the text regions of the image have been detected, the next step is Feature Extraction. In this phase, the features of individual characters are extracted. The performanceof each character recognition system depends on the features that are extracted. The extracted features from the input character should allow the classification of a character in a unique way. Segmentation is done by separation from the individual characters of an image.

After Segmentation and Feature Extraction of the image is done, the final step includes giving the resulting image as input to OCR, which involves the extraction of objects from the image. The OCR function provides an easy way to add text recognition functionality to a wide range of applications.[14]

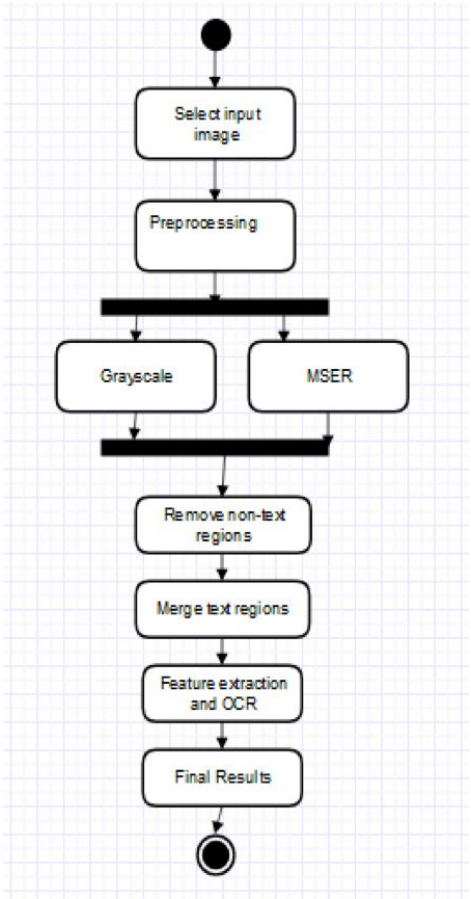

Figure 5: Flowchart for OCR

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

VI. EXPERIMENTAL RESULTS

In order to perform the comparative analysis of the OCR tools, an image consisting of Hindi text is to be given as input to Tesseract as well as the proposed work. The image when given as input to Tesseract gives the output right away on the screen whereas the proposed work shows the transformation of the image at each stage and the output in this case is a text file which contains the identified Hindi text. The results are then compared as to which approach yields desirable results.

PERFORMANCE MEASUREMENT

For the comparison report, a piece of text from a Hindi text book was taken and given as input to both Tesseract and the proposed work. The results obtained were as follows:

Input image:

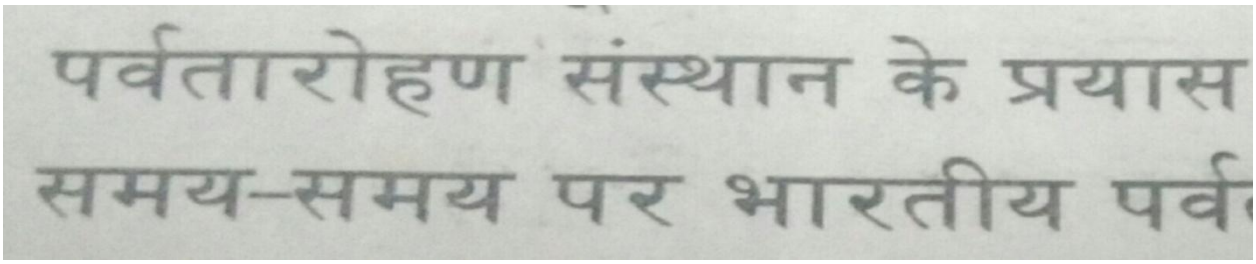

Figure 6 Input Image

Output from Tesseract:

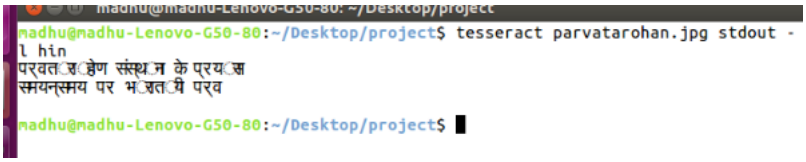

Figure 7 Output from Tesseract for the Input Image

Output from the proposed work at various stages:

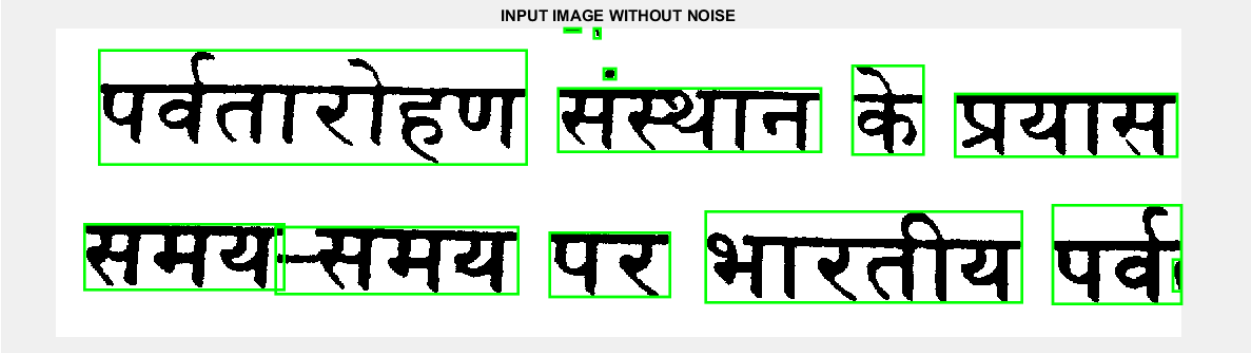

Figure 8 Input image after the noise is removed

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

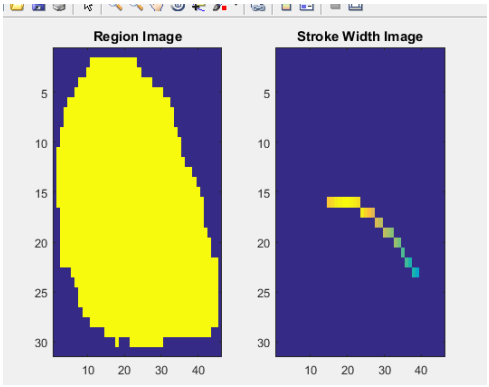

Figure 9Regions of the input image whileStroke width variation is being applied

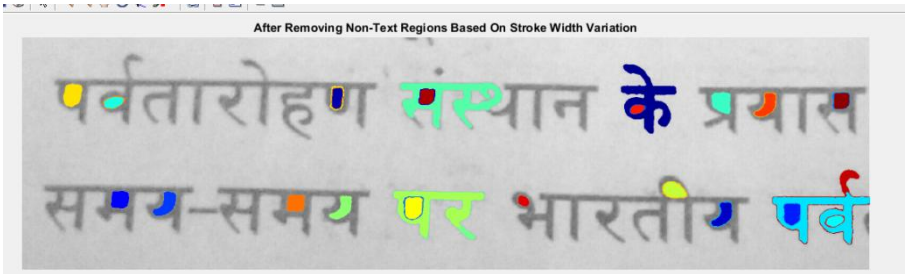

Figure 10Input image afterStroke width variation is applied

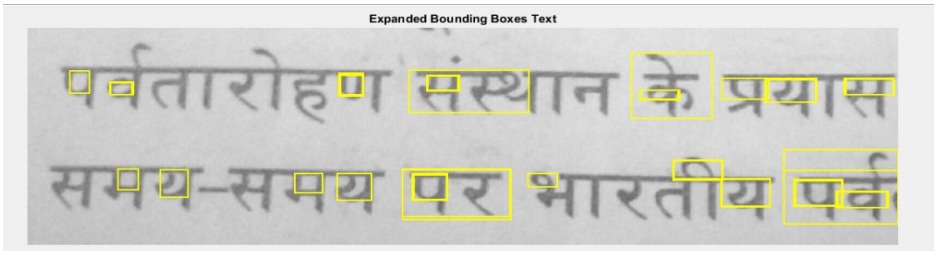

Figure 11Input image when bounding boxes are being created

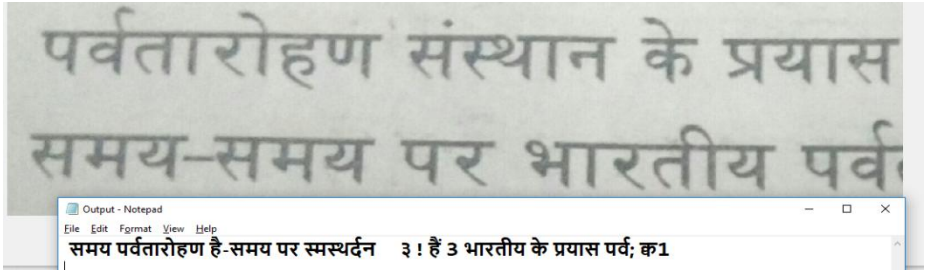

Figure 12 Output being displayed as a text file containing the identified text

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: www.ijirset.com

Vol. 8, Issue 12, December 2019

As we can observe here the output obtained is desirable in the case of proposed work rather than the one obtained from Tesseract. The proposed work shows the transformation of the input image at various stages as and when different techniques are being applied to pre-process the input image and also to better differentiate the text region from the non-text region. The proposed work identifies the half letters involved more accurately than the other. Alsothe number of words identified correctly is comparatively greater in the case of the proposed work.

VII. CONCLUCSIONS AND FUTURE WORK

This work focuses on improving the efficiency of OCR for the Hindi language. It investigates the principles of OCR and the techniques to enhance its performance for the Hindi language implemented in MATLAB. This work will make it possible to have an accurate and readily accessible tool that will help people recognize and understand the language easily and efficiently. This solution to Optical Character Recognition has resultedan improved version of OCR for the Hindi language which carries out various operations on the image before giving it as an input to the OCR. The observation says it is important to separate text regions from the non-text regions so as to get better results and hence techniques like removal of a non-text region using geometrical properties and Stroke Width Variation were used. The concept of bounding boxes which at first divides each word into individual characters and later merges these bounding boxes to form a single bounding box around individual words or text lines is used. This is done to provide words as input to OCR as this enables recognition of the actual words in an image, which carries more meaningful information than what just the individual characters would carry. Finally, feature extraction is carried out and we get the output text. Tesseract does not use the above-mentioned techniques which results in it having lesser accurate results for recognition of Hindi language especially when “maatras”, textbook images, handwritten images and images downloaded from Internet are concerned.

In the future, the implemented algorithms used for the segmentation of Hindi text can be extended further for the recognition of other Indian scripts. Some new features can be added for improving the accuracy of recognition. These algorithms can be tried on a large database of Hindi images. Also, the recognition of digits in the text, half characters, and compound characters can be done to improve the word recognition rate

REFERENCES

[1] GeethanjaliAdlinge, VirendrakumarDhotre, Study of Text Recognition in Images using MSER Approach, S.A.T.I, May 2016.
[2] Neetu Bhatia, study of **“**Optical Character Recognition Techniques**”**, IJARCSSE, May 2014.
[3] MamtaKadyan, DeeptiAhlawat**“**Character Recognition using OCR Algorithm**”**, JNCET, May 2017.
[4] Nitin Mishra, C Patvardhan, C Vasantha Lakshmi, Sarika Singh, study of **“**Shirorekha Chopping Integrated OCR Engine for Enhanced Hindi Language Recognition**”**, IJCA, February 2015.
[5] Arica, N. and Vural, F. T. Y**.** study of **“**An overview of character recognition focused on offline handwriting**”**, IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews,Vol. 31(2), pp. 216-233, 2001.
[6] Automatically Detect and Recognize Text in Natural Images. (2019, September 25). Retrieved from: https://au.mathworks.com/help/vision/examples/automatically-detect-and-recognize-text-in-natural-images.html
[7] Naveen T. S. **“**Word Recognition in Hindi Scripts**”**, December 2014.
[8] Lee L. L. and Gomes, N. R. study of **“**Disconnected handwritten numeral image recognition**”**, Proceedings of the 4th International Conference, ICDAR, pp. 467-470, 2005.
[9] Arora, S.; Bhattacharjee, D.; Nasipuri, M.; Basu, D. K.; and Kundu, M. study of **“**Application of Statistical features in Handwritten Devanagari CharacterRecognition**”**, International Journal of Recent Trends in Engg., Vol. 2(2), pp. 40–42, 2009.
[10] Pratibha Singh, Ajay Verma and Narendra S. Chaudhari study of **“**On the Performance Improvement of Devanagri Handwritten Character Recognition**”**, Hindawi Publishing Corporation, Volume 2015, January 2015.
[11] Ahmed Gari, Mostafa Mrabti, study of **“**Skew Detection and Correction based on Hough Transform and Harris Corner**”**, IEEE May 2017
[12] Bag, S.; Harit, G study of **“**A Survey on Optical Character Recognition for Bangla andDevanagariscripts**”**, Sadhana,Vol.38,pp.133-168,2013.
[13] Bansal, V. and Sinha, R. M. K. **“**Segmentation of touching and fused Devanagari characters**”**, Pattern Recognition, Vol. 35(4), pp. 875-893, 2002.
[14] SahilBadla, study of **“**Improving the Efficiency of Tesseract OCR Engine**”**, SJSU ScholarWorks, May 2014.