Grenze International Journal of Engineering and Technology, June Issue

Devanagari Characters Recognition: Extracting Best Match for Photographed Text

Neelam Chandolikar[1], Swati Shilaskar[2], Vaishali Khupase[3] and Mansi Patil[4]
[1]Department of Information Technology and MCA, Vishwakarma Institute of Technology,Pune, India
Email: neelam.chandolikar@vit.edu
[2-4]Department of Electronics and Telecommunication, Vishwakarma Institute of Technology,Pune, India
Email: swati.shilaskar@vit.edu, vaishali.khupase16@vit.edu, mansi.patil44@gmail.com

***Abstract*—Devanagari script is used by most of the people in India. There are some script-specific structural characteristics of Devanagari script which makes the character recognition problem more challenging. Many OCR tools are available for printed or handwritten Devanagari script recognition. In these systems the input is given in the form of images of the script which can be scanned or photographed. But the existing systems are not robust. They give unexpected results when the input to the system is not ideal that is the image is rotated or tilted or has illumination variance. Our goal is to build a robust OCR system for printed Marathi highlighted text where the variation with respect to font, size, orientation and illumination are allowed. This paper proposes appropriate image transformation techniques to get a robust Devanagari Character Recognition System.**

***Index Terms*— Devanagari, PCA, Tesseract, Levenshtein edit distance, OCR, Perspective transform, Sauvola Thresholding, Highlighted word.**

I. INTRODUCTION

Humans have highly developed sense for several pattern recognition tasks; one such task we very easily perform every now and then is: recognizing the written text. Humans can develop their reading and writing skills in their first few years of education and when they growup they can easily recognize text even if it is printed in different styles, sizes, font and orientation. Even the broken, distorted and misspelled words can also be recognized by human and all this is possible by past experiences. From lots of research it is found that reading skill of computers is still way behind the human. In this paper, the goal is to recognize highlighted Marathi words. An image of highlighted printed or handwritten word is taken as an input. The image can be scanned or photographed using smart-phone or web camera. Many OCR systems for Devanagari script recognition presently exists.

Existing systems have certain limitations like they cannot work on tilted images, images captured at different angles of rotation or in presence of illumination variance. Tilt angle and rotation angle are not the same. The Tilt angle is the angle made by the camera with the plane. Tilt angle is same as the elevation angle whose zero is at the horizon. If it is non-zero, the effects due to tilt are observed in the image and may not give correct results. The rotation angle is the azimuthal angle. Its axis of rotation is perpendicular to the plane. The existing systems may give unexpected result if angle of rotation is non-zero. Also, if there is uneven illumination or shading, the existing systems may fail to recognize the word correctly. So, a methodology is proposed to transform the image

Grenze ID: 01.GIJET.9.2.43

appropriately to nullify the effect due to tilt and the rotation angle. Our system also handles the effect due to variation in background light and uneven illumination to some extent. So, the proposed system achieves desired robustness with respect to above mentioned variations.
The original objective of our OCR system was to provide an appropriate users friendly input method for a Devanagari knowledge search engine. The search engine we are developing is to be used by primary school students. The search queries to be given to the search engine as a photographed image of Devanagari text, taken by smart-phones. This way of searching queries is very easy and user friendly than traditional approach where we have to type Marathi letters from keyboard. Typically, it is quite tedious to input composite Marathi characters via a keyboard. Though main objective of our OCR system was to provide user's friendly input method to our search engine, it can easily be adapted to any other application which requires robust Devanagari character recognition such as Digitization of old documents, texts, Digitization of forms for banks, post offices, or any other government organization, recognition of handwritten name/ amount on cheques, etc.

II. Related Work

Quality of image is very important factor for text recognition. Image quality is degraded due to uneven illumination and when the image is captured in different orientations. Many researchers have developed techniques to improve quality of such degraded images. Authors Huimin Lu et.al worked on shadow removal method for text recognition [1]. It uses binarization of images for better performance. Taeyoung Kim et.al proposed PCA based computation for illumination invariant space. It helps to remove shadow effects from input color image [2]. H. El Bahi et al worked on offline character recognition system for images captured by camera phone.They analyze and compare different thresholding methods to avoid illumination effects. As a result they have chosen Sauvola thresholding method [3]. Sam S. Tsai et.al worked on image matching by using visual text feature of images captured by camera-phone. This work consists of word distance matching method to demonstrate false matches [4].
Annmaria Cherian et.al used Hough transform to correct orientation of perspective input image so that it could recognize text by SVM classifier [5]. Vidula T. V. et. al proposed SURF (Speeded Up Robust Feature) for perspective distorted image, which is faster than SIFT (Scale Invariant Feature Transform) method [6]. J. Sauvola et.al proposed a technique for image binarization. In which they used hybrid approach to adapt defective type images such as change in illumination, noise and resolution[7].Yash Gaurav et.al. presented a Deep Convolutional Neural Network method for classification of inputted images[8]. Shalini Puria et.al proposed Devanagari character classification model their model is based on SVMwhich recognizes printed and handwritten text, they presented some unique preprocessing method for handling shirorekha of Indian scripts[9]. Tripathy et.al presented a SVM based method for Devanagari character recognition using OCR. They provided their work for bangla Devanagari script [10]. Agarwal et.al presented comprehensive survey of methods based on machine Learning Algorithms for Handwritten Devanagari Character Recognition[11]

III. Problem Definition

To build a robust OCR system for printed Marathi highlighted text where the variation with respect to font, size, orientation and illumination are allowed.

IV. Innovative Content

This research offers a unique method to improve performance of OCR. the goal of this work is to recognize highlighted marathi word which is captured by mobile phone, or web camera or any other scanning device. the advantage of this work is that it is rotation invariant, tilt invariant and illumination invariant with maximum accuracy above 98%.

V. Proposed Methodology

In the proposed methodology, we have focused on recognizing Marathi highlighted words in any orientation i.e. tilt or rotation, and also having different illumination effects. For recognizing the highlighted script Tesseract API is used. In case of Marathi Language,Tesseract fails to recognize the script correctly if the tilt and rotation angle are non-zero or have light variance.The paper focuses on improving these Tesseract API limitations. Figure 1 Shows the block diagram of the proposed system, which consists of different phases, beginning with input printed text imagee with a highlighted word, pre-processing, rotation invariant, tilt invariant, illumination

invariant, Tesseractimplementation,finding the best match for the Tesseract output using Levenstein distance and final recognized text. The block diagram of the proposed system as follows:

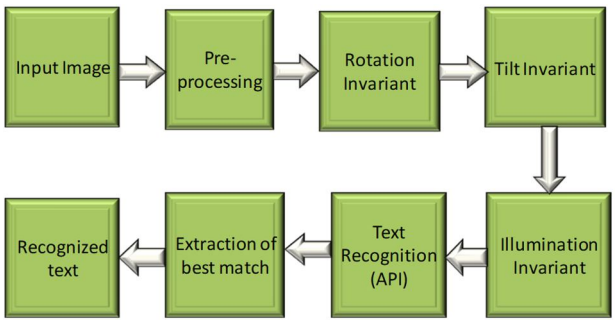

Figure1.Block diagram of proposed system

A. Pre-processing

Pre-processing steps (as shown in Figure2) is applied on the input image to remove the noise from it and also to minimize the variations in the character styles. The scanned document sometimes has Salt and Pepper noise or Shaded areas. This noise must be filtered during preprocessing step. Sometimes image contains some black spots. To remove these black spots and noise along with black shade at the edges, filtering has been done. Here, we have used Median filter to remove high frequency components that cause noise in the image.

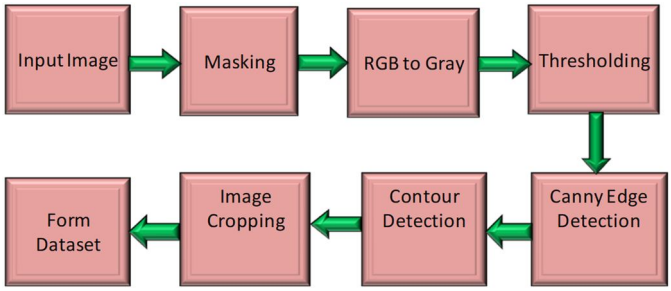

Figure2.Overall flow of Pre-processing

- Input Image: The highlighted printed text is captured by mobile phone, and that captured image is an input to our system. Captured image contains one or more highlighted words. Recognition of these highlighted words is the goal of this project.
- Masking: Masking of highlighted part is done here, for that lower and upper bound of color is found out with the help of BGR values of particular color and then bitwise AND operation is done to extract color part only.
- (RGB)Color image to Grayscale image: The input contains color text image. In preprocessingphase the image is converted to grayscale image.
- Thresholding: Thresholding is also known as binarization. In this certain threshold value has been set, this will convert the pixels to black and white. If pixel value is above the threshold, the pixel is converted into white and if the pixel value is less than threshold value, thepixel is converted into black. Quality of binarized image depends on value of thethreshold.
- Canny Edge Detection: Canny edge detection is a process to extract significant structural information from image and reduce the amount of data to be processed
- Boundary Tracing or Contour Detection: Contour detection means to find out the boundary of the area of interest using edges. It will identify connected components of an image and store that pixel values in the form of array. Contour can be found out by traversing the rows of image which is already filtered. The contour detection algorithm searches foreground pixel and store it into an array by marking it. Similarly, it will find all the neighborhood pixels. This process will continue till all the pixels of the image have been stored or it will continue to search in next row.

B. Rotation Invariance

To make the system rotation invariant, first we need to find the angle by which the image is rotated and then compensate the rotation. This is implemented using step shown in Figure3.

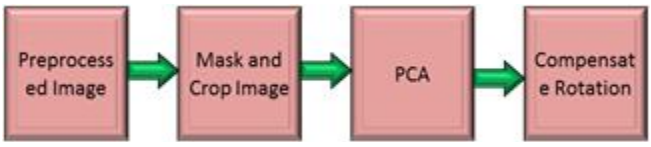

Figure3.Block diagram to make system Rotation Invariant

1. Mask the highlighted portion as already discussed in masking step of pre-processing part.
2. Crop the masked image to get only highlighted portion. This is required since the masked highlighted portion has black background which may give us improper dataset obtained in step 3.
3. Classify the pixels into background pixels and foreground pixels. The black pixels of the text are the foreground pixels and rest others are background pixels.
4. Implement Principal Component Analysis on the foreground pixels obtained in previous step. The brief explanation of PCA is given immediately after step 5.
5. Rotate the image by the negative of the angle obtained in step 4.

Principal Component Analysis (PCA) [12] is used for finding the direction of maximum variance i.e.the directions where the data is most spread out. For finding this direction of maximum variance, Eigen vectors of the covariance matrix associated with the dataset are calculated. The Eigen vector corresponding to largest Eigen value gives the vector in the direction of maximum variance. In this paper, the coordinates of the pixels lying inside the contour of the highlighted word forms the dataset.

Implementation of PCA ,steps are as follows

I. A nx2 matrix of the dataset is formed, where n is the number of pixels lying inside the contour of highlighted pixel.

$$Data = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ x_5 & y_5 \end{bmatrix}$$

II. Find Covariance matrix of the Data matrix:

$$covariance\_matrix = \begin{bmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{bmatrix}$$

Where, $cov(x,y) = \frac{1}{(n-1)}\sum(x_i - \bar{x})(y_i - \bar{y})$ and $var(x) = \frac{1}{(n-1)}\sum(x_i - \bar{x})(x_i - \bar{x})$

III. Find Eigen values and Eigen vectors of the covariance matrix.

IV. Find Eigen vector corresponding to the largest Eigen value. The direction of this Eigen vector can be obtained by

$$angle(in\ radians) = \tan^{-1}\left(\frac{y}{x}\right)$$

The value of angle gives us the angle made by highlighted word with the X axis. If the value is not zero then in order to make it parallel to X axis we need rotate the image by the angle , Rotate angle= 0-angle.

Figure 4 shows Input Image, its HSV Image then Masked Image, followed by Median Filter Image, its Gray Scaled Image,application of Binary Thresholding , Contour Detection, Rotated Image, Median Filter on rotated Image and Final Image.

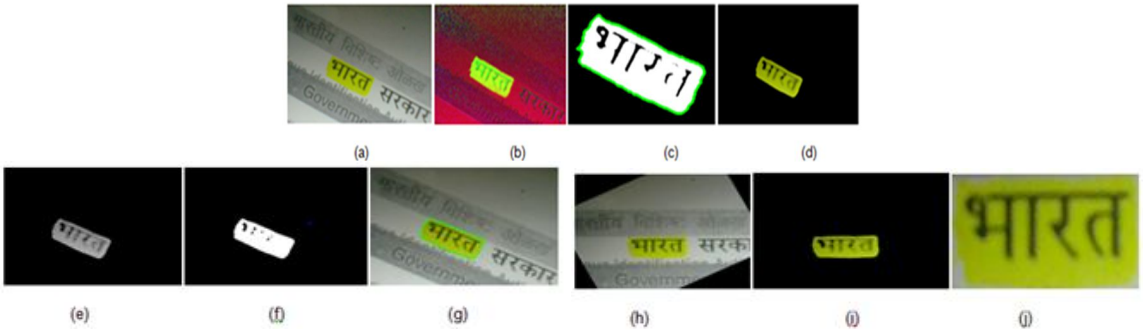

Figure 4.(a)Input Image, (b)HSV Image, (c)Masked Image, (d) Median Filter Image, (e)Gray Scaled Image, (f)Binary Thresholding , (g)Contour Detection, (i)Rotated Image, (j) Median Filter on rotated Image, (k)Final Image

C. Tilt Invariant

When the photos taken at slightly tilted angle, the highlighted word is not visible properly. To make the word properly visible perspective transform method is used.

Implementation of Perspective Transform

To make this system tilt invariant Perspective Transform plays an important role. In the transformed image the letters are **not** slanted and straight. This algorithm is very useful for OCR.

Initially the contour for the highlighted word is found. For perspective transform, we need to define the region of interest which is in the form of rectangle. The coordinates of the vertices of rectangle are such that the top-left point have the smallest (x+y) sum, the bottom right have the largest (x+y) sum, the top-right have smallest (x-y) difference and the bottom-left has largest (x-y) difference. These points are then placed in consistent order. The height and width of the rectangle enclosing the highlighted word can be determined using the above obtained vertices.The first point is (0, 0) in the list of points it indicates the top-left corner. The second point is top-right corner given by (maxWidth - 1, 0), (maxWidth - 1, maxHeight - 1) gives the bottom-right corner and (0, maxHeight - 1) gives the bottom-left corner. In a consistent ordering representation these points are defined.

Top-down view of the image is obtained using cv2.getPerspectiveTransform function. It requires two arguments rectand dst. The rect is the list of four regions of interested points in the original image and dst is list of transformed points. The cv2.getPerspectiveTransform function returns the actual transformation matrix M. The transformation matrix is applied in cv2.warpPerspective function. The transform matrix M, image, height and width of output image pass in to cv2.warpPerspective function resulted into warped image, which is our top-down view. Figure 5 shows steps of making tilt invariant.

(k) (l) (m)

Figure5. (k) Tilted Image, (l) ROI image shown by rectangle, (m) Warped Image

D. Illumination Invariant

To make this system illumination invariant Sauvola thresholding method is used.

Sauvola thresholding

Sauvola thresholding is a local thresholding technique. This technique is useful for text recognition where the background of images is not uniform.[7] In this method thresholds are calculated for every pixel by using formula which is mentioned below. The formula contains the mean and standard deviation of the local neighborhood which is defined by a window centered around the pixel. The local thresholding value will be calculated by the following equation:

$$T(x,y) = m(x,y).\left(1 - k.\left(1 - \frac{s(x,y)}{R}\right)\right)$$

Where k is a constant equal to 0.5, and R denotes the dynamic range of the standard deviation s (defined as R 128 for a grayscale documents)

Algorithm:

input_image

def_mean_std(image[ndarray(N, M)], int(window_size))

{

m=mean of each pixel of image

s=standard deviation of each pixel of image

returns:m, s

}

def_mean_std (image[ndarray(N, M)], int window_size, k=(float), r=None)

{ $T(x,y) = m(x,y).\left(1 - k.\left(1 - \frac{s(x,y)}{R}\right)\right)$

returns:T: [ndarray(N, M)]

}

Above function is used by Sauvola threshold, in which mean and standard deviation of each pixel of an image has been calculated and return by using neighborhood. Here, neighborhood is defined by rectangular window having size w*w. Where, window_size(w) should be odd integer value such as (3, 5, 7, ...). Here, Parameter *window_size* determines the size of the window that contains the surrounding pixels. Here, Sauvola thresholding is applied to an array, threshold value T is calculated using the formula given in algorithm. Where, m(x, y) is mean of pixel (x, y), s(x, y) is standard deviation of pixel (x, y), k is used to weights the effect of standard deviation, R is maximum standard deviation of grayscale image.
This algorithm is used to compensate illumination effects of the image even if the image is captured in different light variations, this Sauvola threshold preserved information contained in an image. Figure 6. has two images, image (n) is the image having illumination effect and image (o) obtained by applying Sauvola thresholding which is very useful for OCR system to recognize text correctly.

(n) (o)

Figure6.(n) Image having illumination effect, (o) Sauvola Threshold image

E. Text Recognition

Text recognition is the most important task of any OCR system, there are various OCR systems are available but they are not capable to produce correct output if there is a variation with respect to rotation, tilt and illumination. So aim of this project to make a robust system where variations with respect to rotation, tilt and illumination are allowed. In this work main focus is on these three aspects.
This text recognition system is implemented using Google API called Tesseract which supports more than 110 languages[13], where Long Short Term Memory neural network is used to train text file. It is used to convert image into text. It has 98% accuracy but when there are variations found in rotation, tilt effects, light effects its accuracy starts decreasing. So, we tried to overcome this problems using PCA to make system rotation invariant, perspective transform to make system tilt invariant and Sauvola thresholding to make system illumination invariant. So recognition rate get increases above 98%.

F. Extraction of Best match

The goal of this work is to make robust OCR system which recognizes the word even if it is distorted, word is misspelled or some characters in the words are deleted. It should be recognized in its correct form and it is done by finding best match to the infected word. For this, thelist of words is stored in text file. The word obtained from previous step is search in that text file, and if there is a match found for that particular word, that best match will be treated as a recognized text. This will increase the accuracy of our system.
To find out best match for the detected word, Levenshtein distance method is used. The Levenshtein distance [14] is a string distance measurement technique. It is used for measuring the difference between two sequences of string. In casual way, the Levenshtein distance between two strings is the minimum number of single-character edits i.e. insertions, deletions or substitutions which are required to change one string into the other. This Levenshtein edit distance method would help us in matching a word or string in its infected form with its original form.
The Levenshtein edit distance method is implemented in following order

- Minimum length of the two words
- Actual Levenshtein edit distance between the words
- Length of subset string match, starting from the first letter

In this project Levenshtein distance method is used to find best match to the recognized Marathi word from the Marathi keywords to increase the accuracy. The keywords list is already stored in the text file. The recognized words found in infected form it will get correct by using Levenstein distance edit method as shown in **Error! Reference source not found.**
Mathematically, the Levenshtein distance between two strings a, b is given by $lev_{(a,b)}$

$$lev_{(a,b)} = \begin{cases} max(i,j) & \text{if } \min(i,j) = 0 \\ min \begin{cases} lev_{(a,b)} = (i-1,j) + 1 \\ lev_{(a,b)} = (i,j-1) + 1 \\ lev_{(a,b)} = (i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases}$$

TABLE I. LEVENSTEIN EDIT DISTANCE OBSERVATION

Incorrect Word	Correct Word
अशुद	अशुद्धी
विद्रयता	विद्रव्यता
नैसरिक	नैसवगिक
अनधा्य	अन्नधान्य

Where $1_{(a\neq b)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1otherwise, and $lev_{a,b}(i,j)$ is the distance between the first i characters of $a$and the first $j$characters of b.
Note that the first element in the minimum corresponds to deletion from (a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same.

VI. PRACTICAL RESULTS AND ANALYSIS

In this section the current result is presented. Below figure show the output of image containing highlighted Marathi printed text '**भारत**'.System becomes rotation invariant by implementing PCA on correct dataset. Even if image is rotated during scanning or capturing the system maintains its accuracy. In this work if the recognized word is sometimes infected, Levenshtein edit distance method is used to extract word which is the best match to the infected word from list which is stored in text file or Marathi dictionary.Implementation of Perspective Transform makes the system tilt invariant. Also the effect of uneven illumination is removed using the Sauvola algorithm.
The Perspective Transform and Sauvola Algorithm work efficiently and give good result. The practical results of each stage are shown in following figures. Figure7 shows rotation invariant result, Figure8 shows tilt invariant result and Figure9 shows illumination invariant result. The final recognized word is shown by Figure10.

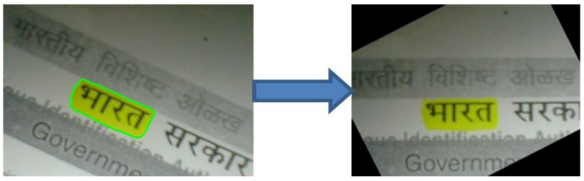

Figure 7.Rotation Invariant Result

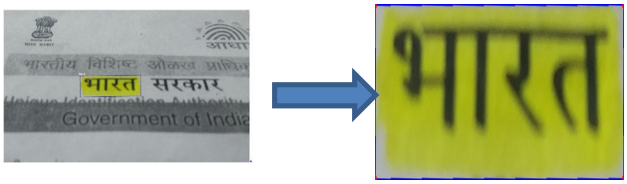

Figure 8.Tilt Invariant Result

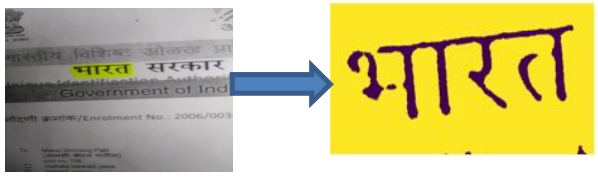

Figure 9.Illumination Invariant Result

Fig.10.Recognized Marathi Text

The experiment is done on 100 images. The experimental analysis shows that the proposed method work fine with long word i.e. word having more than two letters it gives 98% accuracy, the accuracy get reduced due to illumination effects on image.

VII. CONCLUSION

In the context of the current project we are planning to use the character recognition system for knowledge search engine where users are school children, for whom handling complicated Devanagari keyboard might be difficult. For the children, transliteration facility to provide input Marathi words using English keyboard is difficult. So, we allow user to enter the search query by the captured image in which region of interest is highlighted by marker so that user can acquire more information through the knowledge search engine. We use the character recognition system to recognize the input and convert it into digital form and pass it on to our knowledge search Engine. This feature would enable the users to enter search queries in an easy and user friendly manner.The developed character recognition system has several standalone applications as well, such as digitization of documents, automating several systems in which ability to recognize the text/numbers play a crucial role (e.g. recognizing the amounts written on checks, recognizing the addresses written on the envelopes, recognizing names, addresses, phone numbers written on forms etc.), to list a few. The underlying algorithms and techniques which we aim at developing would be applicable to all these applications in general. This approach can be used in multilingual character recognition as well.

REFERENCES

[1] HuiminLu ,Baofeng Guo, Juntao Liu, Xijun Yan “A Shadow Removal Method for Tesseract Text Recognition”,2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017)

[2] Taeyoung Kim, Yu-Wing Tai, Sung-Eui Yoon “PCA ased Computation of Illumination Invariant Space for Road Detection”,2017 IEEE Winter Conference on Applications of Computer Vision

[3] H. El Bahi, Z. Mahani, A. Zatni and S. Saoud “A robust system for printed and handwritten characterrecognition of images obtained by camera phone”,WSEAS TRANSACTIONS on SIGNAL PROCESSING Volume 11, 2015

[4] Sam S. Tsai, Huizhong Chen, David Chen, Vasu Parameswaran, Radek Grzeszczuk, Bernd Girod “Visual Text Features for Image Matching”, 2012 IEEE International Symposium on Multimedia

[5] Annmaria Cherian, Sebastein “Automatic Localization and Recognition of Persectively Distorted Text in Natural Scene Images”

[6] Vidula T. V.,Vrinda V. Nair “A Robust Performance Evaluation Scheme for Rectification Algorithms in Camera Captured Document Images ”, 2014 ICCSC, Trivandram

[7] J. Sauvola and M. Pietikainen, “Adaptive document image binarization,” Pattern Recognition 33(2), pp. 225-236, 2000. DOI:10.1016/S0031-3203(99)00055-2

[8] Y. Gurav, P. Bhagat, R. Jadhav and S. Sinha, "Devanagari Handwritten Character Recognition using Convolutional Neural Networks," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2020, pp. 1-6, doi: 10.1109/ICECCE49384.2020.9179193.

[9] Shalini Puria,, Satya Prakash Singh, “An efficient Devanagari character classification in printed and handwritten documents using SVM”, www.sciencedirect.com Procedia Computer Science 152 (2019) 111–121

[10] Tripathy, Nilamadhaba, Tapabrata Chakraborti, Mita Nasipuri, and Umapada Pal. "A scale and rotation invariant scheme for multi-oriented character recognition." In 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 4041-4046. IEEE, 2016.
Agrawal, Mimansha, Bhanu Chauhan, and Tanisha Agrawal. "Machine Learning Algorithms for Handwritten Devanagari Character Recognition: A Systematic Review." vol 7 (2022): 1-16.

[11] Ding, Chris, Ding Zhou, Xiaofeng He, and HongyuanZha. "R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization." In Proceedings of the 23rd international conference on Machine learning, pp. 281-288. 2006.

[12] Badla, Sahil. "Improving the efficiency of Tesseract OCR Engine." (2014).

[13] Haldar, Rishin, and Debajyoti Mukhopadhyay. "Levenshtein distance technique in dictionary lookup methods: An improved approach." arXiv preprint arXiv:1101.1232 (2011)