2024 Asian Conference on Intelligent Technologies (ACOIT)
Karnataka, India. Sep 6-7, 2024

Effective Compound Character OCR for Printed Devanagari Script

Ansh Mathur, Harshit Gupta, Prem Prakash Vuppuluri,
Faculty of Engineering, Dayalbagh Educational Institute (Deemed to be University),
Dayalbagh, Agra, Uttar Pradesh-282005, India
mathuransh02@gmail.com

2024 Asian Conference on Intelligent Technologies (ACOIT) | 979-8-3503-7495-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ACOIT62457.2024.10939176

***Abstract* — Devanagari script serves as the foundation for three prominent languages: Hindi, spoken by over 520 million, Marathi by over 83 million, and Nepali by over 14 million individuals. It holds the distinction of being the most widely used Brahmic script globally, employed by countless people for reading and writing in their daily lives. Devanagari has a separate class of characters called compound characters which are the conjunction of two consonants. Recognition of printed OCR has been widely explored, however, the use of compound characters in Devanagari makes the problem significantly challenging. This paper focuses on optical character recognition of these compound characters. The task is quite challenging since several of these compound characters are very similar and hence difficult to distinguish. There are a total of 955 such unique characters in Devanagari. We prepared a dataset of these characters, preprocessed them and trained on a linear SVM model. An independent set of test instances was developed as part of the work. A dictionary was also prepared for these characters and complete OCR was performed on these compound characters. The proposed model was tested extensively on the test instances, and obtained an average accuracy of about 84.5%.**

Index Terms* — *Devanagari, Compound Characters, OCR, preprocessing, SVM, segmentation, edge linking, classification, testing.

I. INTRODUCTION

OCR (Optical Character Recognition) is one of the most popular applications of image processing in present day scenario. The importance of devanagari OCR is due to the widespread usage of devanagari script within the globe across various languages such as Hindi, Marathi, Nepali, and Sanskrit. By focusing on devanagari OCR, researchers and scholars can contribute not only in advancing the field of character recognition but also reducing the language and cultural barriers in this highly interconnected world.

OCR was originally started for English language. The importance of OCR in any other language cannot be overseen. There are several other works which motivate to perform OCR on regional languages. These works include the OCR for Hindi Language by Yadav et al 2013, OCR system for printed Kannada documents by Ashwin et al 2002 [3], complete Bengali OCR by Rehman et al 1998 [12], Marathi OCR by Limkar et al 2020 [13]. Two works done for compound character recognition include handwritten compound character recognition by Deore et al 2021 and the handwritten offline devanagari compound character recognition by Sachdeva and Juhi in 2021. This work includes images of 50 characters. Our work includes a larger and more comprehensive dataset taking all the compound characters into consideration.

This paper talks about performing complete OCR on compound characters in devanagari script. Devanagari script has 36 consonants and 12 vowels [2]. When these 36 consonants are written in conjunction with any of the other 35 consonants, the character is known as compound character. There are a few consonants that are never written in conjunction with other consonants. Thus, there are a total of 955 such unique characters.

The rest of the paper is organized in the following sections: Section II talks the dataset preparation, its source, method and the preprocessing done on it. Section III talks about the experimental work done – the model prepared for ocr and testing the model in two ways, including the various segmentations done. Section IV is the results and discussion section which gives the output obtained and the various challenges faced in the process.

II. DATASET PREPARATION

The most important task was to prepare a dataset of those compound characters which are not easily available. After a bit research and inquiring, all the compound characters were obtained. These obtained characters were in the form of text and the aim was to obtain these characters in image format for proper training of the model.

For this task, an open-source online tool *smallseotools* was used. The tool helped in converting each character from text to images as shown in fig. 2.

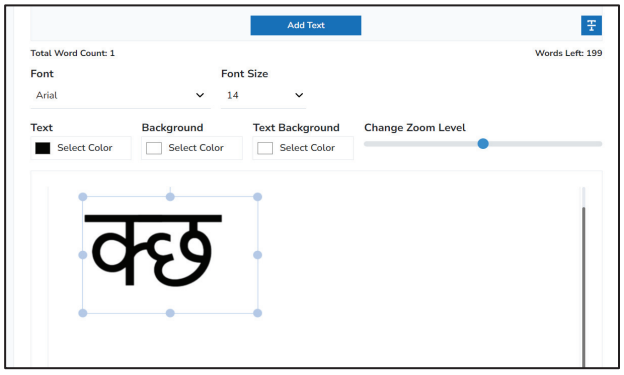

Fig. 1. A snap of the online tool smallseotools showing the text to image generation of character क्ख.

The dataset obtained included images of different sizes with different aspect ratios of the image covered by the characters. To overcome this drawback, the dataset was preprocessed[5] which included the following three steps:

A. Segmentation

The dataset had different size images and different font sizes for the characters. A code was written and run on this dataset which took each image from this dataset, performed segmentation and cropped only that part of image that had the character in it.

B. Thresholding

The image needs to be binarized for better and more accurate training of the linear model like SVM. Each segmented image was then binarized.

C. Resizing

Since the images were cropped according to the size of the characters, they needed to be resized to a same standard size for training. Thus, all these images were resized to 50x50 pixels.

After these three operations on the data, the horizontal line in all the characters were removed. This was done with a special intention of accurate prediction [6]. Later, at the time of prediction, the horizontal line was removed and then given to the model for prediction of label. Thus, this step was necessary. The images were named with their labels from 1 to 955. This new dataset was a preprocessed one and ready to train.

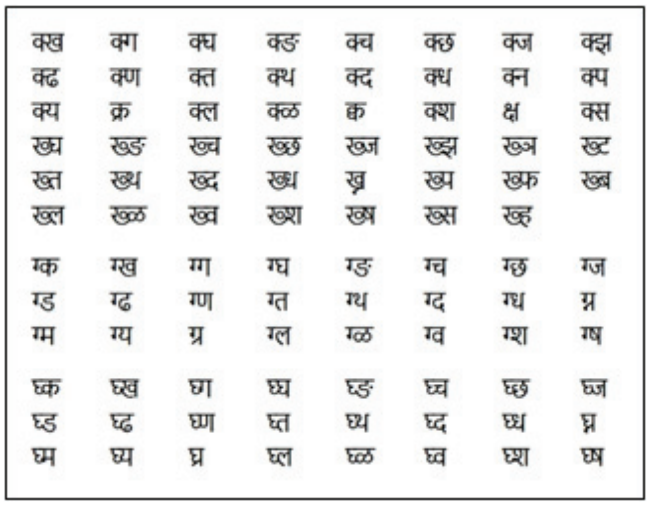

Fig. 2. Compound Characters of Devanagari script.

III. EXPERIMENTAL WORK

A. Training

There are many machine learning models used for the purpose of optical character recognition. Some of them used for devanagari ocr are MLP, SLO, SVM and Simple Logistics. When all the above-mentioned models were compared, SVM proved to be the best in terms of accuracy [7]. Thus, this paper focuses on SVM (Support Vector Machine) model for the purpose of predicting the labels.

A SVM model creates multiple decision boundaries (hyperplanes) to separate multiple classes. Our linear SVM model is trained on 955 different classes where the model creates these hyperplanes in n-dimensional space. SVM model generates a best fit decision boundary in case of two classes.

There was only one image per class. Thus, the data was not divided into testing and training data. Instead, the training was done on the entire data and testing was done on the random 50 images extracted from that training data itself.

B. Testing

The real test of the training was on the text images given as input. The steps followed for that testing are as follows:

- Word Segmentation
- Character Segmentation
- Preprocessing of segmented character
- Predicting the labels
- Output the text using dictionary

1) Image Segmentation

Segmentation includes identifying contours in the entire image. The contours in the entire image were identified and bounding rectangle were drawn to obtain words. Next task includes obtaining characters from these words. From these words, the top horizontal line (head stroke) was removed, and the same segmentation was performed again. This time, detecting contours and drawing bounding rectangles gave us the characters [8].

A small glitch was found in this method. When the head stroke was removed, a few characters were found which were no more connected. This led to formation of two contours within a character and the model treated these as two different characters. To overcome this drawback, the training dataset was targeted. For such characters, the training dataset itself was filled with those two separate contours and even the training considered these as two different characters.

The ultimate goal is to obtain the same text. The model considering it either as a single character or as two different characters and then combining becomes the back-end part. This hardly makes a difference to the user who wants the text as output.

Another flaw faced was the cases where character was joint but due to resizing in pixels, a small gap was generated. Here, edge linking was used to fill up the spaces if the gap is too small [9].

2) Dictionary Preparation

Once a segmented character is obtained, the segmented image needs to be preprocessed in the same manner as the training data was preprocessed. After obtaining the processed image, prediction is done using the pre-trained SVM model. This forms an array of predicted labels. Now, the dictionary comes into consideration. A dictionary was prepared which mapped the labels of images with the corresponding text in devanagari. It is a csv file which has two columns and as many rows as the number of characters in training [10].

The array of predicted labels was read and for each label, corresponding text was picked from the dictionary and displayed on the output screen.

This complete process of testing the developed model is shown diagrammatically in figure 3.

IV. RESULTS AND DISCUSSION

Character segmentation was the most challenging task as devanagari includes a horizontal line at the top which connects all the characters within a word [4]. Secondly, if that horizontal line be removed, there are several characters which are divided into segments within a character. Thus, the techniques used were removing the horizontal line using some algorithm and then perform edge linking to prevent segmentation within characters. These segmented characters formed an array of images, compared to the dataset, predicted using the pretrained model and formed a new array of labels predicted. These labels were then compared to the dictionary prepared and the actual devanagari text was given as the output where the text was same as the one given in the test images. The various compound characters in devanagari are shown in fig. 2. These are a few examples and there are a total of 955 such characters used in this work.

To find the accuracy and correctness of this approach, several test images were given as input, on which complete OCR was performed and accuracy was calculated based on the number of characters correctly predicted out of the total number of characters in that test image. If there is an image with n number of characters and out of them m characters are correctly predicted, then the accuracy is

$$\eta = \frac{m}{n} * 100 \quad ; \; m \leq n \qquad \text{---- (i)}$$

The overall result of this experiment was in two major parts:

- Testing on individual characters
- Output the text

The results of these sections are depicted in figure 5. Since the Optical Character Recognition task performed by us was only for printed text and devanagari text does not have different font styles, so the accuracy on predicting the individual characters was obtained to be 100%. The part (a) of figure 4 shows how it predicted the label of the images of characters picked for testing purpose. First task was successful, and it tick-marked the correctness of training.

Next section deals with segmenting the characters. Section IV(A) of this paper talks about segmentation in detail. After segmentation, these characters were predicted with their labels and this time, the accuracy was not 100%, obviously because of the various reasons such as noise, font size, resizing, thickness, etc. For testing purpose, a text of 140 characters was taken. It segmented the characters as required and predicted them as shown in part (b) of figure 4. A problem faced during this process was the separate parts of characters which were not connected at all but was segmented by the algorithm and considered as a character. These factors led to decrease in accuracy. To overcome this, any segmented character with width less than the threshold was ignored.

The last section of result is about obtaining the actual output required by the user. A user wants a text output of the text displayed in an image. In this part, the labels from the array were compared to the labels column of the dictionary and the corresponding characters were picked from the text column of the dictionary. That text was displayed on the screen accordingly. To ensure that appropriate characters are kept in appropriate words and separated by words, a simple way was used. While segmenting the words, each time the characters segmentation function was called. This function returned the string the which consisted of all the characters of a particular word joined together by a head stroke. Each time this string was added with the previous string along with a space in between. Finally, this complete string was displayed which is the required text. This output is shown in the part (c) of figure 4.

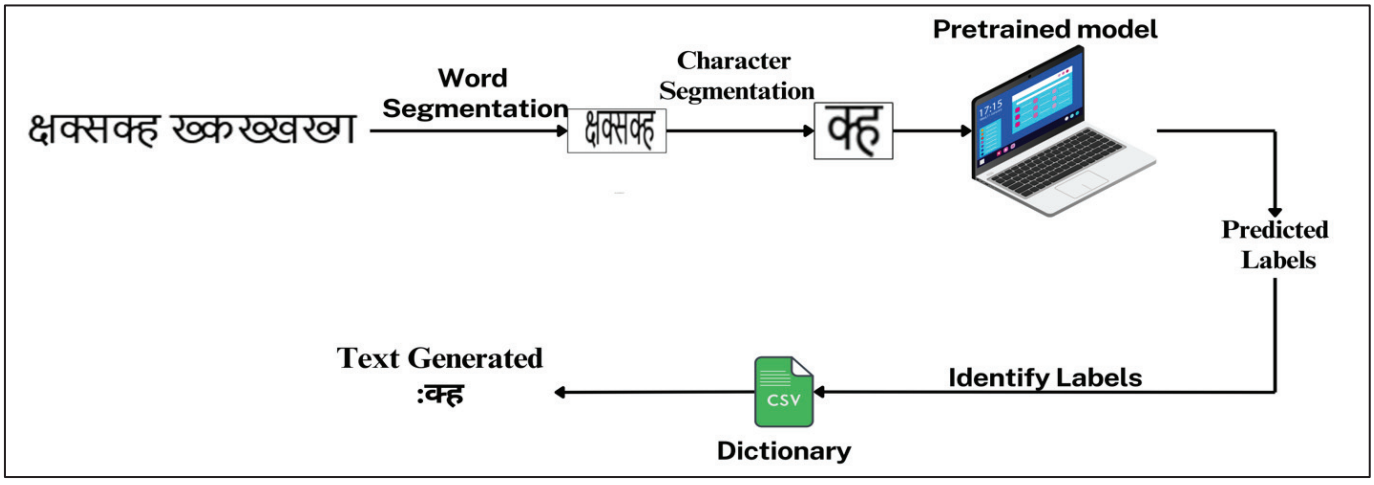

Fig. 3. Flow chart of the model proposed for devanagri compound character OCR

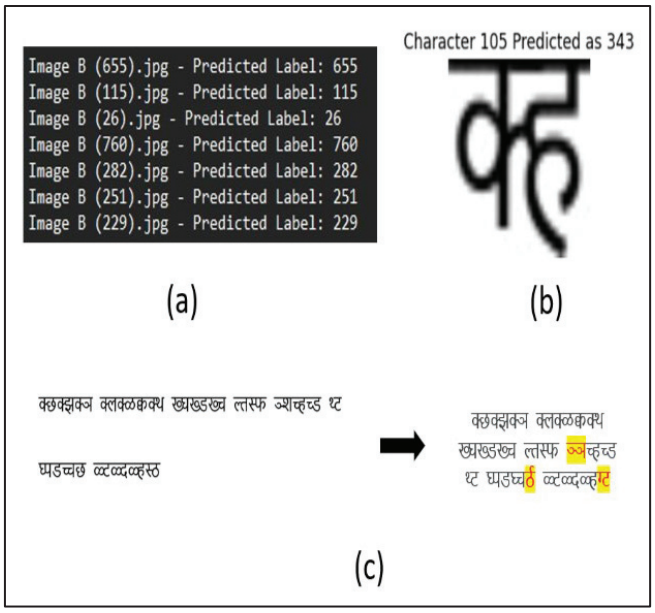

Fig. 4. (a) Prediction of labels for images with individual characters. (b) A segmented character from a complete image of text. (c) The input image and the corresponding text generated by the model.

Out of the various input images, one example is shown in figure 4. As shown, there are a total of 24 characters in the image which were segmented and predicted. On comparing, it was found that 3 characters were wrong and not matching with the actual character as given in the image. Thus, according to the equation (i), the accuracy comes out to be

$$\eta = \frac{(24-3)}{24} * 100 = \frac{21}{24} * 100 = 87.5\ \%$$

Similarly, several other test images. Table no. 1 shows the character count and accuracy for the 10 instances. In all the cases, it ensured that the accuracy was more than 80% with an average accuracy of 84.51%.

TABLE I. RESULTS OF THE TEST ON INDEPENDENT DATASET PREPARED ALONG WITH THE CHARACTERS IT CONSISTS OF, NUMBER OF CHARACTERS PREDICTED WRONG AND THE OVERALL ACCURACY OBTAINED FOR EACH TEST INSTANCE.

Test Instance Number	Number of Characters	Wrongly Predicted Characters	Accuracy (%)
1	24	3	87.5
2	45	9	80
3	34	5	85.3
4	42	5	88.1
5	12	1	91.6
6	65	13	80
7	78	17	78.2
8	64	5	92.1
9	56	10	82.1
10	91	18	80.2

V. CONCLUSION

Optical Character Recognition is always a challenging task due to various challenges like the font size, thickness and similarity between characters [1]. When it comes to devanagari, the number of characters are too much. There are 36 consonants, 12 vowels and there are combination of all 36 consonants with all the 12 vowels as well as the combination of 36 consonants with all 35 other consonants. This makes a difficult task to create a complete dataset for training such a huge number of characters. Secondly, there are various characters in hindi-devanagari that are too similar and are distinguishable only by a single dot or a small curve. This challenging task was taken up for the compound characters which form the combination of two consonants.

The dataset prepared was of 955 compound characters which was preprocessed and made ready to train. A linear SVM model was chosen as the OCR was for printed characters. Devanagari OCR is the field which needs to be researched more rapidly. This script is used by various languages such as Hindi, Nepali, Bengali, Marathi, etc. and the population using this language forms a huge fraction of the world population [11].

The accuracy achieved is 85%in worst case. This can be increased further, if some other model may be used for training or cross-validation may be done in SVM model itself. The future work is to compare various models for this task with various feature extraction and edge location techniques. The best possible combination of model and features may give a better accuracy.

In this paper, entire compound character set was covered and considered for the purpose of recognition. To include individual characters, it increases the accuracy and also makes dataset a huge one. This work opens up a huge ground for further research including the comparison of models for this purpose, inclusion of entire characters set of devanagari, including all consonants, vowel, compound characters and diacritics.

It can be concluded that linear SVM can be used satisfactorily for the purpose of optical character recognition of compound characters of devanagari. It gives an accuracy of about 85% and accurately predicts the labels for individual characters.

REFERENCES

[1] Kompalli, Suryaprakash, Sankalp Nayak, Srirangaraj Setlur, and Venu Govindaraju. "Challenges in OCR of Devanagari documents." In Eighth International Conference on Document Analysis and Recognition (ICDAR'05), pp. 327-331. IEEE, 2005.

[2] S. P. Deore, "Devanagari Handwritten Compound Character Recognition Using Various Machine Learning Algorithms," *2021 2nd International Conference on Communication, Computing and Industry 4.0 (C2I4)*, Bangalore, India, 2021, pp. 1-6, doi: 10.1109/C2I454156.2021.9689425.

[3] Ashwin, T. V., and P. S. Sastry. "A font and size-independent OCR system for printed Kannada documents using support vector machines." Sadhana 27 (2002): 35-58.

[4] Kumar, Vijay, and Pankaj K. Sengar. "Segmentation of printed text in devanagari script and gurmukhi script." International Journal of Computer Applications 3, no. 8 (2010): 30-33.

[5] Mursari, Lily Rojabiyati, and Antoni Wibowo. "The effectiveness of image preprocessing on digital handwritten scripts recognition with the implementation of OCR Tesseract." Computer Engineering and Applications Journal 10, no. 3 (2021): 177-186.

[6] Shafait, Faisal, and Thomas M. Breuel. "A simple and effective approach for border noise removal from document images." In 2009 IEEE 13th International Multitopic Conference, pp. 1-5. IEEE, 2009.

[7] Sachdeva, Juhee, and Sonu Mittal. "Handwritten offline devanagari compound character recognition using machine learning." In ACI'21: Workshop on advances in computational intelligence ISIC. 2021.

[8] Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. "Contour detection and hierarchical image segmentation." IEEE transactions on pattern analysis and machine intelligence 33, no. 5 (2010): 898-916.

[9] Ghita, Ovidiu, and Paul F. Whelan. "Computational approach for edge linking." Journal of Electronic Imaging 11, no. 4 (2002): 479-485.

[10] Tsimpiris, Alkiviadis, Dimitrios Varsamis, Charalampos Strouthopoulos, George Pavlidis, and Kiourt Chairi. "Open-source OCR engine integration with Greek dictionary." In Proceedings of the 25th Pan-Hellenic Conference on Informatics, pp. 436-441. 2021.

[11] Sorate, Priyanka, Navjot Kaur Kanwal, and Manish Malhotra. "Comparison of various writing characteristics of Hindiand Marathi languages of Devanagari origin."

[12] Rahman, A. F. R., and M. Kaykobad. "A complete Bengali OCR: A novel hybrid approach to handwritten Bengali character recognition." *Journal of computing and information technology* 6, no. 4 (1998): 395-413.

[13] Limkar, Suresh, Gautami Mudaliar, Sneha Kulkarni, Neha Rathod, Tejasvi Gadakh, and Sanaya Shah. "Optical character recognition for Marathi language using deep convolutional neural network." Journal of University of Shanghai for Science and Technology 22, no. 12 (2020): 1201-1213.

[14] Yadav, Divakar, Sonia Sánchez-Cuadrado, and Jorge Morato. "Optical character recognition for Hindi language using a neural-network approach." Journal of Information Processing Systems 9, no. 1 (2013): 117-140.