Advances in Intelligent Systems and Computing 1045

Ashish Kumar Luhach
Janos Arpad Kosa
Ramesh Chandra Poonia
Xiao-Zhi Gao
Dharm Singh *Editors*

First International Conference on Sustainable Technologies for Computational Intelligence

Proceedings of ICTSCI 2019

Springer

Multi-font Devanagari Text Recognition Using LSTM Neural Networks

Teja Kundaikar and Jyoti D. Pawar

Abstract Current research in OCR is focusing on the effect of multi-font and multi-size text on OCR accuracy. To the best of our knowledge, no study has been carried out to study the effect of multi-fonts and multi-size text on the accuracy of Devanagari OCRs. The most popular Devanagari OCRs in the market today are Tesseract OCR, Indsenz OCR and eAksharayan OCR. In this research work, we have studied the effect of font styles, namely Nakula, Baloo, Dekko, Biryani and Aparajita on these three OCRs. It has been observed that the accuracy of the Devanagari OCRs is dependent on the type of font style in text document images. Hence, we have proposed a multi-font Devanagari OCR (MFD_OCR), text line recognition model using long short-term memory (LSTM) neural networks. We have created training dataset *Multi_Font_Train*, which consists of text document images and its corresponding text file. This consists of each text line in five different font styles, namely Nakula, Baloo, Dekko, Biryani and Aparajita. The test dataset is created using the text from benchmark dataset [1] for each of the font styles as mentioned above, and they are named as *BMT_Nakula*, *BMT_Baloo*, *BMT_Dekko*, *BMT_Biryani* and *BMT_Aparajita* test dataset. On the evaluation of all OCRs, the MFD_OCR showed consistent accuracy across all these test datasets. It obtained comparatively good accuracy for *BMT_Dekko* and *BMT_Biryani* test datasets. On performing detailed error analysis, we noticed that compared to other Devanagari OCRs, the MFD_OCR has consistent, insertion and deletion type of errors, across all test dataset for each font style. The deletion errors are negligible, ranging from 0.8 to 1.4%.

Keywords Multi-font · OCR · Devanagari

T. Kundaikar (✉) · J. D. Pawar
Department of Computer Science and Technology, Goa University, Taleigao, India
e-mail: dcst.teja@unigoa.ac.in

J. D. Pawar
e-mail: jdp@unioa.ac.in

A. K. Luhach et al. (eds.), *First International Conference on Sustainable Technologies for Computational Intelligence*, Advances in Intelligent Systems and Computing 1045,
https://doi.org/10.1007/978-981-15-0029-9_39

1 Introduction

India is country with 22 constitutionally recognized languages and 12 scripts. Devanagari script is officially used by 10 Indian languages. Historically, lot of Indian literature, religious and scientific books are written in this script. Most of these work is not converted into digital form. In case this literature can be made available in digital form, it will enrich the research field of natural language processing for Indian languages. Optical character recognition (OCR) is a research field in which the text is recognized from scanned documents. There are two methods to recognize text from the scanned document. The first is segmentation-based and the second is segmentation-free approach. In the former approach, the characters are extracted from the input document using segmentation techniques, and further, these extracted characters are recognized using neural network techniques. In the segmentation-free approach, deep learning techniques are used to recognize the text in each line. Using these techniques a lot of work is done in OCR for Roman script, and it shows impressive accuracy on variety of documents. Similar techniques have been used to create OCR for Devanagari script, but the accuracy of this OCR is low as compared to OCR for the Roman script. There are three competitive OCRs for Devanagari script, namely Tesseract OCR [2] open-source tool, Indsenz OCR [3] commercial product and eAksharayan OCR [4] freely available on TDIL [5].

Document in the Roman script is written in different font styles like Times New Roman, Calibri and same is true for document written in Devanagari text. However, we use different font style to represent text for various purposes such as invitation cards, formal documents, notices. On using a different font, the look of text differs as shown in Fig. 1.

We were interested to know the performance of OCR on the scanned document having text in Devanagari script in different font styles. We have tested the Devanagari OCRs with the Devanagari text document images in different font styles and observed that there is a difference in the OCR's accuracy for each of the font styles. In this paper, we have worked on multi-font Devanagari OCR. The content of paper is organized into five sections. Section 1 consists of Introduction and followed by Sect. 2, which gives an idea about related work. Section 3 describes the effect of text document images in different font style on accuracy of Devanagari OCRs. And Sect. 4 discusses creating training and test dataset in form of text document images in different font styles, model to recognize text using LSTM and the result. Section 5 concludes the paper.

Fig. 1 Text in different types of fonts style for Devanagari script

2 Related Work

The first efforts towards the recognition of Devanagari characters in printed documents started in 1970 using segmentation-based approaches. Researchers at Indian Institute of Technology, Kanpur, India, developed a syntactic pattern analysis system for the Devanagari script [6]. In the 1990s, Palit and Chaudhuri [7], Pal and Chaudhuri [8] developed the first complete end-to-end OCR system for Devanagari. Although in the 1990s OCR for Devanagari was contained only at the research level, in the early 2000s it took a major leap when Center for Development of Advance Computing (CDAC) India released the first commercial Hindi OCR called "Chitrankan" [9]. Recently, segmentation-free approach was used for word classification in printed Devanagari documents by [10, 11]. BLSTM-based transcription scheme is used to achieve text recognition for the seven Indian languages [12].

Bidirectional long short-term memory (BLSTM) architecture with connectionist temporal classification (CTC) output layer was employed to recognize printed Urdu text. They have evaluated BLSTM networks for two cases: one ignoring the character's shape variations and the second by considering the variations. The recognition error rate at character level for the first case is 5.15% and for the second is 13.6% [13]. Deep convolutional neural network has been used for handwritten Devanagari text [14]. The iterative procedure using segmentation-free OCR was able to reduce the initial character error of about 23% (obtained from segmentation-based OCR) to less than 7% in few iterations [15]. There has been a lot of work for different scripts and little work for text document images in multi-font for same scripts. We have attempted to create multi-font Devanagari OCR (MFD_OCR) for Devanagari text using LSTM neural networks.

3 Effect of Devanagari Text Document in Different Fonts Styles on Devanagari OCRs

3.1 Devanagari Font Styles—An Overview

Font classification is based on different properties such as internal representation, glyph width, installation of fonts, physical existence, encoding and readability. The internal representation consists of either bitmap or vector fonts. In bitmap font, glyphs are represented as a matrix of dots and vector is represented as curves. The vector fonts consist of true type, open type, meta font and postscript. The Devanagari font design model [16] can visually describe any given Devanagari typeface, and it is this model which has been used as a base for developing the tool. The facets of the model consist of tool, grey value, contrast, axis, vertical terminal, built , horizontal terminal, angular terminal, basic hand, turns, counters, inclination, vertical stem, curves, C to C joinery, end knot, beginning loop, end loop, neck join, vertical proportion, horizontal proposition and neck proportion [16].

Fig. 2 Text in Nakula font style

वह कथा या कविता बदल जाती है

Fig. 3 Text in Baloo font style

वह कथा या कविता बदल जाती है

Fig. 4 Text in Dekko font style

वह कथा या कविता बदल जाती है

Fig. 5 Text in Biryani font style

वह कथा या कविता बदल जाती है

Fig. 6 Text in Aparajita font style

वह कथा या कविता बदल जाती है

The style of fonts differs based on the facets chosen to design font style. There are more than 300 fonts for Devanagari script. We have considered different properties such as vertical terminal with serif, horizontal and left swoosh, grey value with light and dark. Considering these properties of the font, we have chosen fonts, namely Nakula, Baloo, Dekko, Biryani and Aparajita, which has regular style. The description for each of it is given below.
Nakula Font: The Nakula Devanagari font, which has been developed by IMRC, India, for the University of Cambridge. The font is Unicode compliant, and the font is TrueType. It contains all the conjuncts and other ligatures (including Vedic accents) likely to be needed by Sanskritists. It follows the rounded glyphs and little thin/thick variation as shown in Fig. 2.
Baloo Font: The Baloo font is Unicode compliant and libre licensed font. It has a distinctive heavy spurless design as shown in Fig. 3 [17].
Dekko Font: The inter-letter spacing of this font is wider, allowing for it to be used at smaller sizes on screens. This fonts stroke contrast has thick horizontals. The pen angles traditionally associated with Devanagari has some diagonal stress, but here the script uses a vertical stress as shown in Fig. 4 [18].
Biryani Font: Biryani is a libre font development project. Its fonts are designed in a monolinear, geometric sans serif style. Like several early geometric sans typefaces from the last century, Biryani font characters have a strong flavour to them. Figure 5 shows Biryani font in regular font style [19].
Aparajita Font: The Aparajita font is high OpenType font and is Unicode-encoded. Figure 6 [20] shows Aparajita font in regular font style.

3.2 Performance of OCRs on Devanagari Text Document Images in Aparajita and Nakula Font Styles

We have used the text from [1] to create text document images in Aparajita and Nakula font style. The same is used to check the performance of the OCRs on text document in different font style.

Table 1 Performance of OCR's character accuracy for text in Aparajita and Nakula font style

Font	OCR		
	Tesseract (%)	Indsenz (%)	eAksharayan (%)
Nakula	**90.14**	81.74	70.14
Aparajita	**77.27**	33.21	22.20

Bold indicates the highest accuracy for each of the fonts

Table 2 Error analysis of OCRs for text document image in Aparajita font styles

Type of error	OCR		
	Tesseract (%)	Indsenz (%)	eAksharayan (%)
Insertion	9.01	5.17	15.07
Substitution	9.54	41.52	55.32
Deletion	3.37	19.34	7.39

Table 3 Error analysis of OCRs for text document image in Nakula font styles

Type of error	OCR		
	Tesseract (%)	Indsenz (%)	eAksharayan (%)
Insertion	4.27	5.59	10.58
Substitution	3.53	7.49	33.19
Deletion	1.86	4.20	8.09

We have studied the performance of the three most popular Devanagari OCRs, namely Tesseract, Indsenz and eAksharayan OCR on text document image in different font styles, namely Nakula and Aparajita. The performance of OCRs is shown in Table 1. The results showed that OCR accuracy is dependent on the text document image in different font styles. We were interested to know the reason for the errors in the OCR, for which it does not achieve 100% accuracy. This made us look at the types of errors it generates. There are three types of errors OCR produces, i.e. insertion, deletion and substitution errors. These number of errors can be understood with the help of OCR evaluation tool [21]. Table 2 shows the percentage of each type of errors for text document images in Aparajita font style, and Table 3 shows the percentage of each type of errors for document images in Nakula font style. We can observe from this that insertion and deletion errors are different for these two font styles. This shows that the type of OCR error differs for text document in different font styles. As there are three types of errors in OCR, i.e. insertion, substitution and deletion. It is difficult to handle the insertion and deletion type of error in post-processing of OCR output. In this work, we have attempted to minimize insertion and deletion type of errors and to create OCR which shows consistent accuracy across all font styles.

4 Proposed Multi-font Devanagari OCR (MFD_OCR) Using LSTM Neural Network

4.1 Data Creation of Text Document Image for Different Font Styles

We have identified this above-mentioned five different fonts to create training dataset using text from corpus available at Centre for Indian Language Technology [22]. We created the images for each line in text corpus for each of the font styles using python packages. The training dataset is created for five fonts, namely Nakula, Baloo, Dekko, Biryani and Aparajita. It consists of an image file of each line in text document images and its corresponding text. Training dataset consists of total 500 K images and the same number of text files. This training dataset is named as *Multi_Font_Train* dataset.

To create test data, we have used text from benchmark dataset IIITH [1], which consists of 100 image files and corresponding text files . The test data is created using following steps:

1. We created text files named as *Text_Files_TestDataset* for each of the text line in these 100 text files of benchmark dataset. There are total 2812 text files in *Text_Files_TestDataset*.
2. For each of the file in *Text_Files_TestDataset*, we have created corresponding text document images for Nakula font style using python script, which take input as text and font style and output is same text as document image in ".png" format. This is named as *BMT_Nakula* dataset, which as 2812 files in ".png" format.
3. Like step 2, we have created test dataset for Baloo font style named as *BMT_Baloo* dataset consisting of 2812 files in ".png" format.
4. Similarly, we have created test dataset for Dekko, Biryani and Aparajita font style named as *BMT_Dekko*, *BMT_Biryani* and *BMT_Aparajita*, respectively.

4.2 Multi-font Devanagari OCR: Text Recognition Using LSTM Neural Networks

OCRopus [23] is an open-source document analysis and recognition tool, and we have used it to create the multi-font Devanagari OCR. Recurrent neural networks (RNNs) are used for recognition, especially 1D LSTM architecture. In this architecture, a window of height equals to that of the input image named as "depth of the sequence" and width equal to 1 pixel called a "frame" and is moved over the text line. These 1D sequences (frames) are then fed to the input layer of the LSTM network. To achieve normalization, it is important to make the depth of sequence equal for all input text line images. We have used an LSTM recurrent neural net to learn this sequence mapping of Devanagari text in different font styles. Our model has 48 inputs, 200 nodes in a hidden layer and 249 outputs. The inputs to the network are columns of pixels. The columns in the image are fed into the network, one at a time, from left to right. The outputs are scores for each possible letter. Learning rate and momentum were set to a standard value which is 0.0001 and 0.9, respectively. It took 800 K iteration, to generate a model with stabilized accuracy.

4.3 Experimental Results

Accuracy of Tesseract, Indsenz, eAksharayan and MFD_OCR on text document images in different font styles.

We have used an abbreviation to each of the OCR as mention here, T_OCR for Tesseract OCR, I_OCR for Indsenz OCR and E_OCR for eAksharayan OCR. ISRI analytic tool is used for the evaluation of OCRs [21]. We have evaluated the performance of the proposed MFD_OCR along with T_OCR, I_OCR and E_OCR with the created test dataset for all five font styles. The performance of OCRs for each of the test dataset is shown in Table 4.

The best character accuracy for *BMT_Nakula* test dataset is shown by T_OCR. Again the accuracy for Text in BMT_Baloo test dataset is greater for T_OCR compared to all other OCRs. The proposed MFD_OCR performs better for *BMT_Dekko* and *BMT_Biryani* test dataset compared to all other OCRs. Finally, Tesseract OCR is more accurate for *BMT_Aparajita* test dataset. The Indsenz and eAksharayan OCRs produce very low accuracy for *BMT_Aparajita* test dataset. It has been observed that each OCR accuracy varies for text document image in different font styles, but MFD_OCR obtained consistent accuracy across all the text document in different font styles.

Error analysis for T_OCR, I_OCR, E_OCR and MFD_OCR for each of the test dataset.

We have also performed the error analysis of the OCRs with the help of OCR evaluation tool [21]. The error in OCR is due to insertion, substitution and deletion of the character. We have discussed these type of errors for each of the test dataset.

Table 4 OCRs character accuracy on test dataset

Dataset	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
BMT_Nakula	**90.14**	81.74	70.14	89.23
BMT_Baloo	**89.64**	78.15	62.93	88.49
BMT_Dekko	77.87	72.34	47.51	**89.03**
BMT_Biryani	88.56	77.28	59.41	**88.89**
BMT_Aparajita	**77.27**	33.21	22.20	69.87

Bold indicates the highest character accuracy for each of the test datasets

Table 5 Percentage of errors of OCRs for *BMT_Nakula* test dataset

Type of error	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
Insertion	**4.27**	5.59	10.58	4.91
Substitution	**3.53**	7.49	33.19	5.55
Deletion	1.86	4.20	8.09	**0.86**

Bold indicates the lowest error rate

Table 6 Percentage of errors of OCRs for *BMT_Baloo* test dataset

Type of error	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
Insertion	4.98	7.78	20.19	**4.58**
Substitution	**2.72**	8.46	14.24	5.39
Deletion	2.10	3.90	2.62	**0.90**

Bold indicates the lowest error rate

Table 7 Percentage of errors of OCRs for *BMT_Dekko* test dataset

Type of error	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
Insertion	6.05	7.07	6.50	**2.23**
Substitution	7.71	12.60	30.35	**6.79**
Deletion	7.86	7.19	15.62	**1.49**

Bold indicates the lowest error rate

Table 5 shows the percentage of each error type for above-mentioned OCRs for *BMT_Nakula* test dataset. The insertion and substitution types of error are less in T_OCR for *BMT_Nakula* dataset. The MFD_OCR has less deletion type of errors, and insertion type of error is nearly the same as that of T_OCR.

Table 6 shows the percentage of errors for each of the OCR for *BMT_Baloo* dataset. The insertion and deletion type of errors are lowest in MFD_OCR for the *BMT_Baloo* dataset. The T_OCR has less substitution type of errors.

Table 7 shows the percentage of errors for each of the OCR for *BMT_Dekko* dataset. The insertion, substitution and deletion type of error are less in MFD_OCR for *BMT_Dekko* dataset.

Table 8 shows that the percentage of errors for each of the OCR for *BMT_Biryani* test dataset. The T_OCR has less substitution and insertion type of errors. The MFD_OCR has less deletion type of errors and insertion, and substitution type of error is nearly the same as that of T_OCR.

Table 9 shows that the percentage of errors for each of the OCR for *BMT_Aparajita* test dataset. The insertion type of error is less in I_OCR for *BMT_Aparajita* dataset. The MFD_OCR has less deletion type of errors, and The T_OCR has less substitution type of errors.

Table 8 Percentage of errors of OCRs for *BMT_Biryani* test dataset

Type of error	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
Insertion	**4.29**	5.63	4.56	4.42
Substitution	**4.09**	10.75	23.37	4.94
Deletion	2.54	5.22	12.65	**1.00**

Bold indicates the lowest error rate

Table 9 Percentage of errors of OCRs for *BMT_Aparajita* test dataset

Type of error	OCR			
	T_OCR (%)	I_OCR (%)	E_OCR (%)	MFD_OCR (%)
Insertion	9.01	**5.17**	15.07	6.24
Substitution	**9.54**	41.52	55.32	22.61
Deletion	3.37	19.34	7.39	**1.40**

Bold indicates the lowest error rate

Percentage of types of errors for each of the OCRs for text document image in different font styles.

Here we have shown the plot of the percentage of each type of errors for all OCRs across all the test datasets as shown below. Figure 7 is a plot of the percentage of deletion type of errors for each of the test dataset versus OCRs.

It is observed from Fig. 7 that the proposed MFD_OCR gives fewer deletion errors for all types of font, which has been tested with a particular type of dataset.

Figure 8 is a plot of the percentage of insertion type of errors for each of the test dataset versus OCRs. Figure 8 shows MFD_OCR has consistent insertion type of error for all fonts, which has been tested with a particular type of dataset.

Figure 9 shows that the substitution types of error are also consistent for all fonts in MFD_OCR except the text document in Aparajita font style.

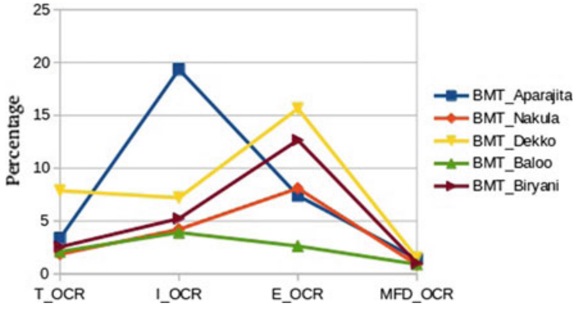

Fig. 7 Deletion type of errors of OCRs for text document in different fonts

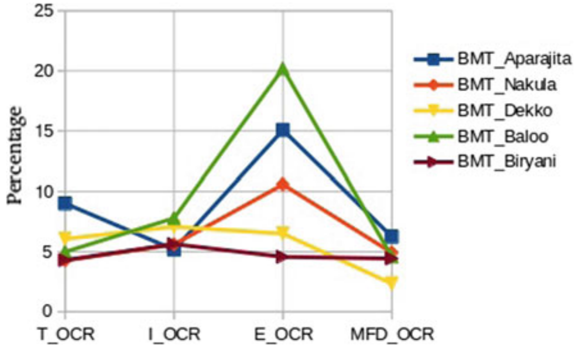

Fig. 8 Insertion type of errors of OCRs for text document in different fonts

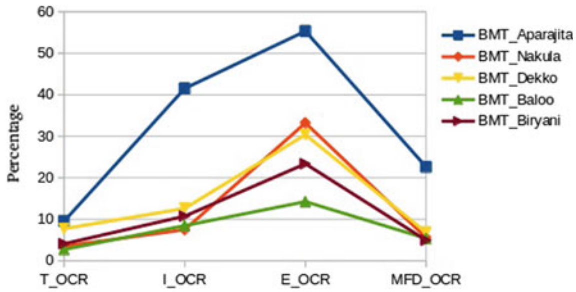

Fig. 9 Substitution type of errors of OCRs for text document in different fonts

5 Conclusion and Future Scope

In this paper, initially, we carried out the study of the effect of text in different fonts style on Devanagari OCRs. We also analysed the percentage of different types of error for which the accuracy of OCRs was low. We observed that the most popular OCR's accuracy is different for text in different font style. The Tesseract OCR obtained good accuracy among all OCRs but showed the highest insertion and deletion type of errors compared to the substitution type of errors. In post-processing of OCR output, it is difficult to handle insertion and deletion type of errors.

We implemented multi-font Devanagari OCR(MFD_OCR) using LSTM neural networks and *Multi_Font_Train* dataset. On the evaluation of MFD_OCR, it showed consistent accuracy, across all the text document images in different font style. It obtained comparatively good accuracy for text document images in Dekko and Biryani font style. On performing detailed error analysis, it was noticed that compared to other Devanagari OCRs, the insertion and deletion type of errors are consistent for MFD_OCR across all text document images in different font style. The deletion errors are negligible, ranging from 0.8 to 1.4%. The Tesseract OCR has the lowest substitution type of errors compared to all OCRs. The accuracy of Tesseract is highest for three test datasets, namely *BMT_Nakula*, *BMT_Baloo* and *BMT_Aparajita*. Using the proposed MFD_OCR, we were able to minimize the insertion and deletion type of errors. These type of errors are difficult to handle in the post-processing stage. We are currently working on the possibility of further improving the accuracy by understanding the substitution types of error.

References

1. Mathew, M., Singh, A.K., Jawahar, C.: Multilingual OCR for indic scripts. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 186–191 (2016)
2. Source OCR: Tesseract OCR [Computer software manual]. Retrieved from https://github.com/tesseract-ocr/. Accessed 14 Aug 2018
3. Hellwig, O.: Indsenz ocr [Computer software manual]. Retrieved from http://www.indsenz.com/int/index.php. Accessed 14 Aug 2018
4. tdil: eaksharaya [Computer software manual] (2018). Retrieved from http://tdil-dc.in/eocr/index.html. Accessed 14 Aug 2018
5. Technology development for Indian languages [Computer software manual]. Retrieved from http://tdil.meity.gov.in/. Accessed 26 Jan 2019
6. Sinha, R.: A syntactic pattern analysis system and its application to Devanagari script recognition. Electrical Engineering Department (1973)
7. Palit, S., Chaudhuri, B.: A feature-based scheme for the machine recognition of printed Devanagari script. In: Das, P.P., Chatterjee, B.N. (eds.) Pattern Recognition, Image Processing and Computer Vision, pp. 163–168. Narosa Publishing House, New Delhi, India
8. Pal, U., Chaudhuri, B.: Printed Devanagari script ocr system. VIVEK-BOMBAY **10**, 12–24 (1997)
9. Pal, U., Chaudhuri, B.: Indian script character recognition: a survey. Pattern Recogn. **37**(9), 1887–1899 (2004)
10. Karayil, T., Ul-Hasan, A., Breuel, T.M.: A segmentation-free approach for printed Devanagari script recognition. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 946–950 (2015)
11. Sankaran, N., Jawahar, C.: Recognition of printed Devanagari text using BLSTM neural network. In: ICPR, vol. 12, pp. 322–325 (2012)
12. Krishnan, P., Sankaran, N., Singh, A.K., Jawahar, C.: Towards a robust OCR system for indic scripts. In: 2014 11th IAPR International Workshop on Document Analysis Systems (DAS), pp. 141–145 (2014)
13. Ul-Hasan, A., Ahmed, S.B., Rashid, F., Shafait, F., Breuel, T.M.: Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks. In: 2013 12th international conference on Document Analysis and Recognition (ICDAR), pp. 1061–1065 (2013)
14. Acharya, S., Pant, A.K., Gyawali, P.K.: Deep learning based large scale handwritten Devanagari character recognition. In: 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), pp. 1–6 (2015)
15. Ul-Hasan, A., Bukhari, S.S., Dengel, A.: Ocroract: A sequence learning OCR system trained on isolated characters. In: DAS, pp. 174–179 (2016)
16. Dalvi, G.: Terminology of Devanagari typefaces [Computer software manual]. Retrieved from http://dsource.in/tool/devft/en/terminology.php. Accessed 20 Jan 2019
17. Type, D.E.: Baloo fonts [Computer software manual]. Retrieved from https://fonts.google.com/specimen/Baloo. Accessed 24 Jan 2019
18. Type, D.S.: Dekko fonts [Computer software manual]. Retrieved from https://fonts.google.com/specimen/Dekko. Accessed 24 Jan 2019
19. Font Biryani, G.: Biryani fonts [Computer software manual]. Retrieved from https://fonts.google.com/specimen/Biryani. Accessed 24 Jan 2019
20. Microsoft Corporation: Aparajita fonts [Computer software manual]. Retrieved from https://catalog.monotype.com/font/microsoft-corporation/aparajita/regular. Accessed 24 Jan 2019

21. Rice, S.V., Nartker, T.A.: The ISRI analytic tools for OCR evaluation. UNLV/Information Science Research Institute, TR-96-02 (1996)
22. cfilt.iitb: Word frequency for Hindi [Computer software manual]. Retrieved from http://www.cfilt.iitb.ac.in/Downloads.html. Accessed 14 Aug 2018
23. OCRopy: Open source document analysis and OCR system. [Computer software manual]. Retrieved from https://github.com/tmbdev/ocropy. Accessed 14 Jan 2019