International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 3, June, 2024, pp. 909-915
Journal homepage: http://iieta.org/journals/isi

Modi Document Transcription to Devanagari

Teja Kundaikar*, Swapnil Fadte, Ramdas Karmali, Ramrao Wagh, Jyoti D. Pawar

Discipline of Computer Science & Technology, Goa Business School, Goa University, Goa 403206, India

Corresponding Author Email: teja.kundaikar@unigoa.ac.in

https://doi.org/10.18280/isi.290311

Received: 4 September 2023
Revised: 24 May 2024
Accepted: 30 May 2024
Available online: 20 June 2024

Keywords:
Modi, Devanagari, Long Short-Term Memory (LSTM), Optical Character Recognition (OCR), transcription, dataset, script

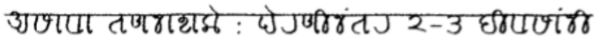
ABSTRACT

Modi [moːɖiː] being ancient script that is not on the list of recognized official scripts for Indian languages; relatively little research has been done to identify handwritten characters in *Modi* compared to other Indian scripts. Character recognition in *Modi* script can be difficult because of the cursive, continuous, unconstrained, and numerous strikingly similar shapes of the characters. Other difficulties in the *Modi* character identification process are segmentation, noise and degradation, the presence of various skews, variations in illumination, uneven alignment, slanting lines, overlapping lines, and contacting lines. Word segmentation or recognition is ineffective for *Modi* script documents because they do not have any word or sentence ending symbols like other scripts. Another problem is the unavailability of a dataset covering most of the syllables required to automate transcription of *Modi* documents. The previous work reported on automatic *Modi* character recognition is on *Modi* characters dataset, i.e. vowels, consonants and numerals. The dataset used for recognition of characters is handwritten characters. This work did not include consonants with vowel diacritic and conjunct consonants. In 2020 the Word Transcription of *Modi* script to Devanagari was reported, which considered only 57 character classes in *Modi*. However, 57 classes are too few to capture the script's characters. We require a dataset that includes vowels, consonants, each consonant with the vowel diacritics and conjunct consonants to cover a wide variety of syllables in *Modi*. This demands looking at different *Modi* document recognition approaches and making them available in widely known scripts such as the Devanagari script. This paper presents a model to recognize the *Modi* text from an input image and make its transcription available in the Devanagari script. In this work, we have also created a dataset that includes *Modi* vowels, consonants, numerals, consonants with vowel diacritic and conjunct consonants. The dataset created consists of text in *Modi* and its transcription in Devanagari. Our proposed model (*ModiDev_LSTM_Model*) for *Modi* documents transcription to the Devanagari using LSTM Neural Networks showed an encouraging character accuracy of 94.67%. Detailed analysis of substitution errors made by the *ModiDev_LSTM_Model*, showed that there are seven types of error, namely *'Anusuvar' (Bindu), 'Eekar', 'Ookar', 'Ardhacandra', 'Matra', 'Aa' and 'other'*. Among these, the highest percentage of substitution error was shown by *'Anusuvar'*, and the lowest was the *'Aa'* error type.

1. INTRODUCTION

There exists a lot of cultural and historical literature in the *Modi* script for the Marathi language. *Modi* script was used during the period of the Maratha empire, so many official [1] and historical [2] documents are archived in *Modi* script. Among several theories about the origin of this script, one of them claims that Hemadpant (or Hemadri Pandit) had developed the *Modi* script during the reign of Mahadev Yadav and Ramdev Yadav (1260–1309 CE). At that time, writing in Devanagari was time-consuming due to the lifting of hand after every stroke. Later use of printing technology for *Modi* was unavailable; as a result, the Devanagari script was declared as an official script for writing in Marathi language. [3]. India has extensive collections of *Modi* records that have been preserved. Bharat Itihas Sanshodhan Mandal in Pune, Tanjavur's Saraswati Mahal, Rajwade Sanshodhan Mandal, and Dhule (Maharashtra) are known to have such collections of documents [4]. Large amounts of *Modi* documents are being cataloged and managed at multiple libraries concurrently with the attempt to revive the script. Script Encoding Initiative (SEI) [5] has completed the project to encode *Modi* and make it available in Unicode. To access and understand the cultural and historical literature available in *Modi*, there is a need to make it available in popular scripts, such as Devanagari. Figure 1 shows the *Modi* script and its corresponding transcription in the Devanagari. One method to make this possible is with the help of experts who understand *Modi* and can make it available in Devanagari script. However, this can be time-consuming and subject to the availability of human experts. Hence, there is a definite need for automatic transcription of *Modi* resources to the Devanagari script. However, this can be challenging due to the characters' cursive, continuous, unrestricted, and several strikingly similar shapes. Segmentation is a challenging step in the *Modi* character identification process. Significant problems include noise and degradation, the presence of different skews, fluctuations in light, uneven alignment, slanting lines, overlapping lines, and contacting lines. Also, they lack the word and sentence ending symbols found in other scripts. The unavailability of datasets for automation is another issue. This paper presents a work that recognizes the *Modi* text from an input image and makes its transcription available in the Devanagari. We have also discussed the creation of a synthetic dataset consisting of *Modi* document images and their transcriptions in Devanagari.

The paper is organized as follows. Section 2 provides details of *Modi* character set and its Devanagari equivalent characters. The related work done for the *Modi* character recognition, *Modi* transliteration software and *Modi* image transcription work are discussed in Section 3. The methodology for our experiment is presented in Section 4. Experimental results and analysis are discussed in Section 5. Finally, the conclusion and future scope are in Section 6.



𑘀𑘭𑘰𑘪𑘰 𑘝𑘜𑘡𑘰𑘫𑘎𑘹 : 𑘢𑘹𑘨𑘜𑘲𑘡𑘽𑘝𑘨 𑙒-𑙓 𑘟𑘱𑘪𑘭𑘰𑘽𑘡𑘲

(a) text in *Modi* script using MarathiCursive font [1]

असावा तणनाशके: पेरणीनंतर २-३ दिवसांनी

(b) Devanagari transcription of *Modi* text.

Figure 1. Text in *Modi* script and its corresponding transliterated text in Devanagari script

2. MODI CHARACTER SET AND ITS EQUIVALENT IN DEVANAGARI AND ROMAN

Table 1. Vowels and vowel diacritics (signs): (*Modi* and Devanagari)

Independent vowels in *Modi*	𑘀	𑘁	𑘂	𑘃	𑘄	𑘅	𑘆	𑘇	𑘈	𑘉	𑘊	𑘋	𑘌	𑘍
Dependent vowel signs in *Modi*	--	◌𑘰	◌𑘱	◌𑘲	◌𑘳	◌𑘴	◌𑘵	◌𑘶	◌𑘷	◌𑘸	◌𑘹	◌𑘺	◌𑘻	◌𑘼
Devanagari script equivalent	अ	आ	इ	ई	उ	ऊ	ऋ	ॠ	ऌ	ॡ	ए	ऐ	ओ	औ
Roman script equivalent	a	aa	i	ee	u	oo	ru	roo	lu	loo	e	ai	o	au
Phonetic equivalent	[ə]	[a]	[i]	[iː]	[u]	[uː]	[ru]	[ruː]	[lu]	[luː]	[e]	[əi]	[o]	[əu]

Table 2. Consonants: (*Modi* and Devanagari)

Consonatat in *Modi* script	𑘎	𑘏	𑘐	𑘑	𑘒	𑘓	𑘔	𑘕	𑘖	𑘗	𑘘	𑘙
Devanagari script equivalent	क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ
Roman script equivalent	ka	kha	ga	gha	ṅa	ca	cha	ja	jha	ña	ṭa	ṭha
Phonetic equivalent	[kə]	[kʰə]	[gə]	[gʱə]	[ŋə]	[tsə]	[tsʰə]	[dzə]	[dzʱə]	[jə̃]	[ʈə]	[ʈʰə]
Consonant in *Modi* script	𑘚	𑘛	𑘜	𑘝	𑘞	𑘟	𑘠	𑘡	𑘢	𑘣	𑘤	𑘥
Devanagari script equivalent	ड	ढ	ण	त	थ	द	ध	न	प	फ	ब	भ
Roman script equivalent	ḍa	ḍha	ṇa	ta	tha	da	dha	na	pa	pha	ba	bha
Phonetic equivalent	[ɖə]	[ɖʱə]	[ɳə]	[tə]	[tʰə]	[də]	[dʱə]	[nə]	[pə]	[pʰə/fə]	[bə]	[bʱə]
Consonant in *Modi* script	𑘦	𑘧	𑘨	𑘩	𑘪	𑘫	𑘬	𑘭	𑘮	𑘯		
Devanagari script equivalent	म	य	र	ल	व	श	ष	स	ह	ळ		
Roman script equivalent	ma	ya	ra	la	va	śa	ṣa	sa	ha	ḷa		
Phonetic equivalent	[mə]	[jə]	[rə]	[lə]	[ʋə]	[ʃə]	[ʂə]	[sə]	[ɦə]	[ɭə]		

Table 3. Numerals: (*Modi* and Devanagari)

Modi numeral	𑙐	𑙑	𑙒	𑙓	𑙔	𑙕	𑙖	𑙗	𑙘	𑙙
Devanagari script equivalent	०	१	२	३	४	५	६	७	८	९
Roman Script equivalent	0	1	2	3	4	5	6	7	8	9

Table 4. Consonants with some vowel diacritics

Consonant with vowel diacritic in *Modi* script	𑘎	𑘎𑘰	𑘎𑘲	𑘎𑘳	𑘎𑘹	𑘎𑘺	𑘎𑘻	𑘎𑘼
Devanagari script equivalent	क	का	कि	कु	के	कै	को	कौ
Roman script equivalent	ka	ka	ki	ku	ke	kai	ko	kau
Phonetic equivalent	[kə]	[ka]	[ki]	[ku]	[ke]	[kəi]	[ko]	[kau]
Consonant with vowel diacritic in *Modi* script	𑘑	𑘑𑘰	𑘑𑘲	𑘑𑘳	𑘑𑘹	𑘑 ◌𑘺	𑘑𑘻	𑘑𑘼
Devanagari script equivalent	घ	घा	घि	घु	घे	घै	घो	घौ
Roman script equivalent	Gha	Gha	Ghi	Ghu	Ghe	Ghai	Gho	Ghau
Phonetic equivalent	[gʰə]	[gʰa]	[gʰi]	[gʰu]	[gʰe]	[gʰəi]	[gʰo]	[gʰau]

The *Modi* script has 48 distinct characters: 14 vowels and 34 consonants [6]. Apart from this, numerals and character modifiers include 10 digits and 14 vowel diacritics. Vowel diacritic can be combined with consonants to form 14*34=476 unique combinations. Consonants can be combined with 476 combinations to get conjunct consonants (Jodakshars), giving syllables a distinct appearance. Theoretically, all syllables can be combined with consonants to get complex Jodakshars, which is rare in practice. *Modi* also has character modifiers like *Anusuwara* (◌◌̇), *Visarga* (◌◌:) and *Ardhacandra* (◌◌̆), which can further modify the appearance and pronunciation of syllables/characters. All of these characters/syllables form the character set for the script. Some of the character set of *Modi* have been provided in the tables. Table 1 shows vowels and vowel diacritics in *Modi* and their transcription in other scripts. Similarly, Table 2 shows consonants. Table 3 shows *Modi* numerals and their transcriptions in Devanagari and Roman script. Table 4 shows a few consonants with vowel diacritic. *Modi* has different letter forms and rendering behaviors. Certain consonants, vowels, and numerals share similar shapes with Devanagari, but the actual difference can be seen in how these characters behave when consonant-vowel combinations and conjunct consonants appear in a text [7-9]. This is a unique feature of *Modi* script, which is not seen in Devanagari script. Consonants with vowel diacritic are shown in Table 4.

3. RELATED WORK

This section is organized into three parts. In the first part, we discussed previous *Modi* character recognition work. *Modi* text transliteration softwares are listed in the second part, and Transcriptions of *Modi* images to Devanagari in the third part.

3.1 *Modi* character recognition

Compared to other Indian scripts, very little work has been done to recognize handwritten characters in *Modi* because it is an old script and is not on the list of official scripts for Indian languages. A review of all the techniques used to recognize handwritten characters and numerals written in *Modi* is reported [10, 11]. These reviews grouped the recognition work into two categories: (i) numeral recognition and (ii) character recognition (i.e., vowel and consonant).

In 2011 Besekar classified the numerals written in *Modi* script using morphology and Decision tree [12]. Besekar and Ramteke also classified numerals written in *Modi* script using Four square zones, variance, theta angle, Rh distance, and variance [13]. However, this work does not include the *Modi* character set. Further, in 2012, Besekar classified the vowels written in *Modi* script using a two-layer feedforward network with the scaled conjugate gradient [14]. However, this work recognizes only *Modi* vowels. Anam and Gupta in 2015, recognized characters written in *Modi* using Kohonen Neural Network with a character accuracy of 72.6% [15]. However, they considered only 22 handwritten characters written in one handwriting style. Thus, it reports work for some subset of *Modi* characters having character accuracy for each character ranging from 85% to 90%. The work by Joseph et al. [16] reports 91.75% accuracy using Daubechies wavelet (Db2) for *Modi* script character recognition. In 2021, Joseph and George [17] used CNN to recognize 46 *Modi* characters, which showed 99.78% accuracy. In 2020, work on the word Transcription of *Modi* script to Devanagari using a Deep Neural Network has been reported, and they have considered 57 classes of the *Modi* script characters [18]. Also, in their work, it is mentioned that more character classes need to be considered. The literature review indicates that little published work is available on *Modi* character recognition, and no standard dataset is available [17]. However, these all work on *Modi* character, i.e. vowels, consonants and numerals, which does not include *Modi* consonants with vowel diacritic.

3.2 *Modi* text transliteration softwares

Modi script transliteration software is created by C-DAC (Centre for Development of Advanced Computing) Mumbai, India, which can convert digital text written in Devanagari script to *Modi* and vice-versa through a website [8]. Aksharamukha Script Converter [18] transliterates text from one script to another, which also includes *Modi*. Moreover, it features text recognition from an image that does not support *Modi* script as input.

3.3 Transcription of *Modi* image to Devanagari

Most of the work is reported on vowel consonants and numerals recognition in *Modi* script. Only one work exists on *Modi* to Devanagari's word transcription, achieving 95.97% accuracy using CNN for 57 classes of *Modi* character set [19], which only includes some consonants with vowel diacritics and conjunct consonant combinations. The challenges for automatic *Modi* transcription are as follows:

·The characters have curves and no punctuation marks.

·There is no space between the words. Continuous sentences are written without any space, hence it is challenging to segment words from the actual *Modi* documents.

·Similar looking characters such as ' 𑘕 '([dzə]) and '𑘡'([nə]), '𑘢 '([pə]) and '𑘮'([hə]) as shown in Table 2.

·Several syllables that have more than one representation [7-9].

·*Modi* script dataset is unavailable, covering all the cases mentioned above.

In this paper, we created a dataset with *Modi* characters, i.e., vowels, consonants, numerals, consonants with vowel diacritic and conjunct consonants, as discussed in section 4.1. Further, we have created a model that automatically transcripts text from *Modi* image documents and makes it available in Devanagari.

4. METHODOLOGY

This section, provides the dataset and model creation methodology. We have created the test and training dataset, which is discussed below in Section 4.1. The procedure to create a model to recognize *Modi* script and transcribe it to Devanagari script using LSTM Neural Network has been discussed in Section 4.2.

4.1 Dataset creation

Since no standard dataset is available for *Modi* that covers all the character set, thus we created a dataset that covers *Modi* consonants, vowels, numerals, consonants with vowel diacritic and conjunct consonants. The process adopted for creating the training and test dataset is the same as mentioned below. Due to the unavailability of the standard *Modi* to Devanagari Transcription dataset, we have created a synthetic dataset for this work using data augmentation. This dataset consists of *Modi* text as an image, i.e. “.png” format, and its corresponding Devanagari Transcription in “.txt” format. The properties of the dataset are presented in Table 5.

Table 5. Dataset characteristics

Type of Data	Number of Text Lines/Sentences	Training Data	Test Data
Raw Data *	9932	7945	1987
Dataset ”Modi images”	99320	79450	19870

***Source**: Marathi General Text Corpus, TDIL [20] and Marathi corpus from IIT-Bombay [21], both from Agriculture domain

The procedure below has been used to create the dataset.

1. To create a training dataset, we took Marathi text corpus in the Devanagari script and its characteristics are given in Table 5. This data consists of vowels, consonants, conjunct consonants and consonants with diacritics. We split it in an 80:20 ratio for training and testing, named as *“train_mar.txt”* and *“test_mar.txt”*, respectively.

2. To generate the training dataset, we gave *“train_mar.txt”* and ’Marathi CursiveT’ font, which is *Modi* script font that makes use of the Devanagari OpenType feature. This renders Devanagari text into *Modi* text. Thus it shows *Modi* script instead of Devanagari [22], which is given as input to below in step 3.

3. To generate a synthetic dataset, we have created an image having text written in *Modi* using ImageDraw, which takes text and appropriate font for the text (Pillow, ImageFont, Python Imaging Library https://pypi.org/project/Pillow/ Image-) and creates text image of the same. Then this image is augmented by two ways: i) Applying a Gaussian filter with a sigma value from 0.5 to 0.7, randomly chosen, and utilizing affine transformation for interpolation. ii) Applying a Gaussian filter with a sigma value 10.0 and a geometric transformation. An example for both the cases is shown in Table 6, which is supported by ocropus-linegen [23]; Please note that the sigma values and the number of images to be generated are changes as per our need.

4. Thus, we have generated ten images for each text line in the text file *“train_mar.txt”*. The output were 79450 image files in *Modi* script in “.png” format at 300dpi resolution and corresponding text in Devanagari. This training dataset was named as *ModiDev_MarathiCursiveT_train* dataset. Similarly, we created a test dataset *ModiDev_MarathiCursiveT_test* using “test_mar.txt”. The details are mentioned in Table 5.

A sample image from the dataset and its Devanagari transcription is shown in Figure 1. An example of two different images in the dataset of the same text is shown in Table 6. One with geometric transformation and a Gaussian filter with sigma 10. Another with affine transformation, Gaussian filter with sigma 0.5. These are done to make the recognition model robust for end documents with *Modi* text affected by document background removal. Moreover, there may be a tilt in writing, as the *Modi* document writing is not in a straight horizontal line.

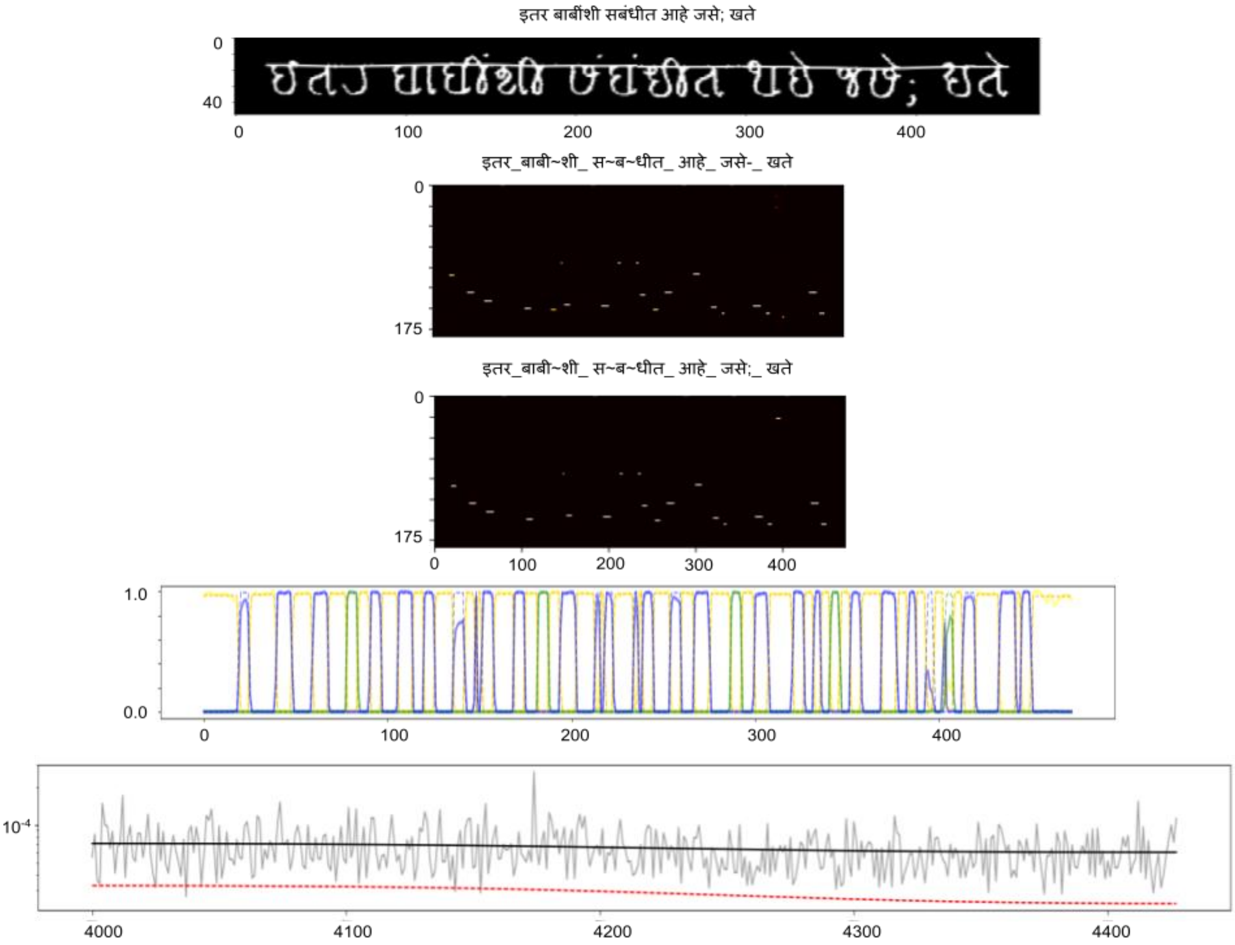


Figure 2. Model building iteration

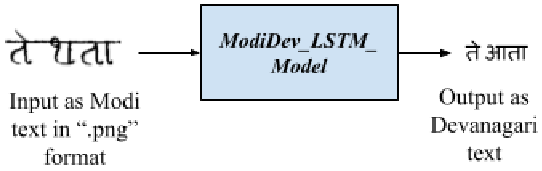
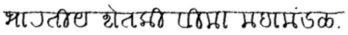
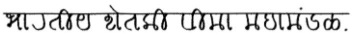

ते थाता
ModiDev_LSTM_ Model
ते आता
Input as Modi text in ".png" format
Output as Devanagari text

Figure 3. *ModiDev_LSTM_Model* transcription output

Table 6. A sample images from a dataset

Actual text	भारतीय शेतकी विमा महामंडळ.
Modi text image (Geometric transformation, Gaussian filter with sigma 10)	
Modi text image (Affine transformation, Gaussian filter with sigma 0.5)	

4.2 Experimental setup to create *Modi* to Devanagari script model using LSTM neural network

The *ModiDev_LSTM_Model* is created using LSTM Neural Network from OCRopus [23], which is a python-based tool for documents. Here we have changed the hidden and output nodes of LSTM Neural Network to achieve maximum accuracy. The purpose of this model is to transliterate the *Modi* images to the Devanagari text. The details of the model creation process are provided in the following steps:

(1) The LSTM Neural Network architecture has initialized as 1D LSTM with 48 input nodes, 200 hidden nodes, and 249 output nodes.

(2) The input frame is set as the 1X height of input image dimensions.

(3) Learning rate and momentum were set to a standard value which is 0.0001 and 0.9, respectively.

(4) The rest of the features were used as default in OCRopus [23].

(5) The *ModiDev_Notosans_train* dataset created in section 4.1 is the input to the LSTM Neural Network. The input to the network was columns of pixels. The columns in the input image are fed into the LSTM Neural Network, one at a time, from left to right.

(6) The final model was created in 25000 steps. The error rate decreases for each step (refer to the red dotted line at the bottom in Figure 2).

(7) The output of this process is a model that transliterates the *Modi* text in ".png" format to corresponding Devanagari text, and we have named this model as *ModiDev_LSTM_Mode*l. as shown in Figure 3.

5. EXPERIMENTAL RESULTS AND ANALYSIS

The Performance of *ModiDev_LSTM_Mode*l and experimental results are presented in Section 5.1, and detailed error analysis is presented in Section 5.2.

5.1 Performance of *ModiDev_LSTM_Model* on *ModiDev_MarathiCursiveT_test* dataset

The *ModiDev_LSTM_Model* is used to transcribe the *Modi* text in image to Devanagari editable text, as shown in Figure 3. We have evaluated the *ModiDev_LSTM_Model* with *ModiDev_MarathiCursiveT_test* dataset using the ISRI analytic tool [24]. It showed 94.67% character accuracy on the *ModiDev_MarathiCursiveT_test* dataset. This is promising accuracy compared to work on Word transcription of *Modi* script to Devanagari Using Deep Neural Network [24], which showed 95.97% accuracy on *Modi* characters, which only included character set with 57 classes. The confusion matrix for errors is shown in Figure 4. From the confusion matrix, we can observe that the errors are due to incorrect recognition of Anusuvar (Anusuvar is a Modi sign ANUSVARA.[25]), Eekar (Eekar is a Dependent vowel signs I in Modi.[25]), Ookar (Ookar is a Dependent vowel signs U in Modi.[25]), Matra (Matra is a Dependent vowel signs E in Modi.[25]), Ardhacandra (Ardhacandra is a Modi sign ARDHACANDRA.[25]), etc.

Table 7. Correct image in *Modi* script and transcription text by *ModiDev_LSTM_Model*

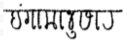
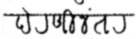
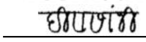
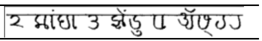
***Modi* text in image**	***Actual Devanagari Transcription***	***ModiDev_LSTM_Model* transcription output**
	हंगामानुसार	हगामानुसार
	पेरणीनंतरं	पेर”णनतर
	दिवसांनी	िदवसानी

Table 7 shows an example of an error that occurred, which has three columns. The first column has the text in *Modi* script, the second column displays the correct Devanagari transcription and the *ModiDev_LSTM_Model* transcription is in the third column. Table 8 shows an example of *Modi* text, its actual transcription and corresponding *ModiDev_LSTM_Model.* From Table 7 and Table 8 we can observe that there are different types of errors, such as Anusuvar, Ookar, Eekar, Ardhacandra, Aa, Matra, and other characters that were transcribed wrongly by *ModiDev_LSTM_Model*. Further, we studied the percentage of each of the errors in detail and categorized them into error classes, which is discussed in Section 5.2.

Table 8. An example of *Modi* images, its actual transcription and *ModiDev_LSTM_Model* output

Modi text image input	
Actual text	२ कांदा ३ झेंडू व अ◌◌ॅस्टर
ModiDev_LSTM_Model output	२ कादा ३ झेडूव एस्टर

5.2 Analysis of errors

The following steps were followed to do a detailed study on substitution errors made by ModiDev_LSTM_Model:

1. The ModiDev_MarathiCursiveT_test image files were given as input to *ModiDev_LSTM_Model*, which produced text files with the text in Devanagari script. The output files are named as *ModiDev_transcript_text* files.

2. The text files from the *ModiDev_MarathiCursiveT_test* dataset and *ModiDev_transcript_tex*t files were aligned using the Recursive Alignment Tool [26]. The output generated from this tool consists of the correct word and transcribed word by our model, which were saved in a file aligned_files.

3. The words are split into syllables. Further, we have categorized the error into seven classes.

The seven error classes are as follows:

Anusuvar: In this class, syllable transcription is correct, except the *ModiDev_LSTM_Model* failed to identify the *Anusuvar* modifier. The percentage of error in this class is the highest among all error classes, i.e. 2.25%. An example is shown in Table 9, where the word कांदा (onion) is transcribed as कादा.

Eekar: In this class, the syllable transcription is correct, except the *ModiDev_LSTM_Model* failed to identify the Eekar modifier. The error percentage is the second highest among all error classes i.e. 1.56%.

Ookar: In this class, the syllable transcription is correct, except the *ModiDev_LSTM_Model* failed to identify the Ookar modifier, which has 0.48% errors.

Ardhacandra: In this class, the *ModiDev_LSTM_Model* failed to identify the Ardhacandra modifier, which has 0.61% errors.

Matra: In this class, the *ModiDev_LSTM_Model* failed to identify the Matra modifier, which has 0.21% errors.

Aa: In this class, the *ModiDev_LSTM_Model* recognized extra Aa, which has 0.07% errors.

Other: In this class, the syllable transcriptions are those that are not from all the above error classes, which have 0.15% errors.

Table 9 provides sample images for different categories of errors. It also shows the expected actual output, incorrect character in red, and category of error/s.

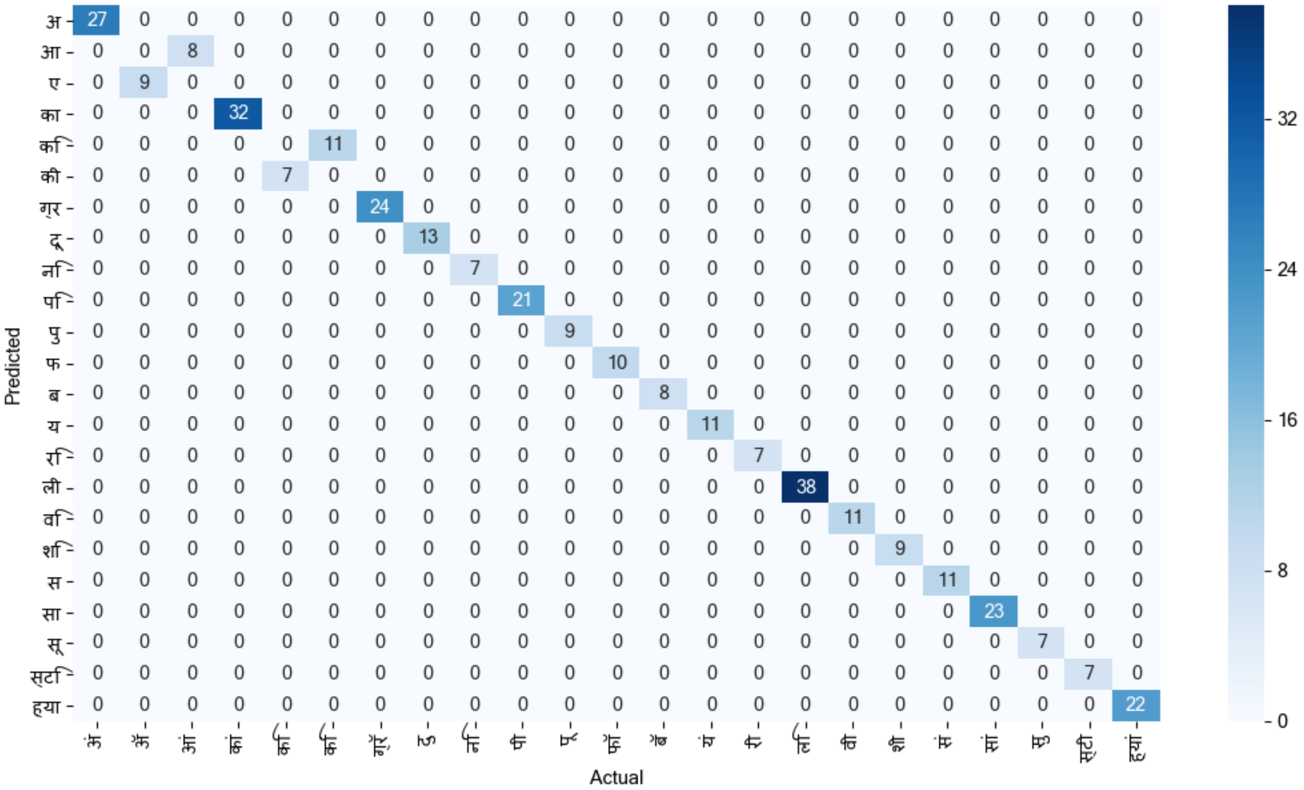
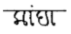
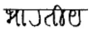
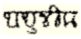
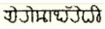
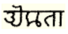
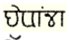
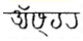
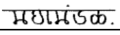


Figure 4. Confusion matrix for substitution errors

Table 9. Example of errors

Sr No	Input Image	Actual Devanagari Transcription	ModiDev_LSTM_Model Transcription Output	Error Class
1		कांदा	कादा	A
2		भारतीय	भार>तय	B
3		आधुनिक	आध्नीक	B and C
4		एरोमाथॅरेपी	एरोमाथ~रेपी	D
5		ऐकता	~कता	E
6		देवांना	देवा~ा	A and F
7		अॅॅस्टर	एस्टर	D and F
8		महामंडळ	महामडह	A and G

6. CONCLUSION AND FUTURE SCOPE

This paper presents a dataset created for *Modi* script with *Modi* text with vowels, consonants, numerals, consonants with vowel diacritic, and conjunct consonants, making it more comprehensive for *Modi* script. Moreover, images are created with a slight tilt, as the *Modi* document writing is not in a straight horizontal line. It presents a model for automatically transcribing *Modi* script images to Devanagari text, showing 94.67% character accuracy on the *Modi* test dataset. However, the model's capabilities are limited to processing low-resolution images when it employs a Gaussian filter with sigma values ranging from 0.5 to 0.7. We are interested in working on handwritten *Modi* manuscripts for Devanagari in the future. However, the handwritten characters are in different styles; thus, we must consider more *Modi* fonts and font-size (whose writing is similar to handwritten data) to handle *Modi* manuscript data.

ACKNOWLEDGEMENT

We are grateful for Mr. Somnath Perni for providing Modi books. We also thank Mr. Narayan Bandodkar and Mr. Tejas Lotlikar for exploring the Modi font.

REFERENCES

[1] C-DAC. (2016). Modi-Script Tools. https://cdac.in/index.aspx?id=lu_modi_script.

[2] Sohoni, P. (2017). Marathi of a single type: The demise of the Modi script. Modern Asian Studies, 51(3): 662-685. https://doi.org/10.1017/S0026749X15000542

[3] Reviving Modi-script. (2014). Typography and Culture. https://www.typoday.in/2014/spk_papers14/rajendrathakre-typo14.pdf.

[4] Khillari, R. (2008). Modi script History. http://modiscript.blogspot.com/2008/05/history-of-modi-script.html.

[5] Berkeley- Linguistics. (2018). Script Encoding. https://scriptencodinginitiative.github.io/.

[6] Simon Ager. (2013). Online Encyclopedia of writing systems and language. Omniglot. https://omniglot.com/writing/modi.htm.

[7] Kulkarni, M.R. (2011). Tumhich Modi Shika (Marathi). Diamond Publications Pune, 264/3 Shaniawar Peth.

[8] Khilari, R. (2010). Learn MoDi script. http://modi-script.blogspot.com/2010/05/learn-modi-script-online.html.

[9] Rajendra Bhimraoji Thakre. (2016). Reviving Modi-Script. MIT Institute of Design. https://shivajiscript.blogspot.com/2016/07/.

[10] Joseph, S., George, J. (2019). Feature extraction and classification techniques of MODI script character recognition. Pertanika Journal of Science & Technology, 27(4): 1649-1669.

[11] Kulkarni Sadanand, A., Borde Prashant, L., Manza Ramesh, R., Yannawar Pravin, L. (2015). Review on recent advances in automatic handwritten MODI script recognition. International Journal of Computer Applications, 115(19): 5-12.

[12] Besekar, D N. (2011). Recognition of numerals of MODI script using morphological approach. International Referred Research Journal.

[13] Besekar, D.N., Ramteke, R.J. (2012). Feature extraction algorithm for handwritten numerals recognition of MODI script using zoning-based approach. International Journal of Systems, Algorithms & Applications, 2: 1-4.

[14] Besekar, D.N. (2012). Special approach for recognition of handwritten MODI Script's vowels. International Journal of Computer Applications (IJCA), 48: 52.

[15] Anam, S., Gupta, S. (2015). An approach for recognizing Modi Lipi using Otsu's Binarization algorithm and kohenen neural network. International Journal of Computer Applications, 111(2): 28-34. https://doi.org/10.5120/19511-1128

[16] Joseph, S., Datta, A., Anto, O., Philip, S., George, J. (2021). OCR system framework for MODI scripts using data augmentation and convolutional neural network. In Data Science and Security: Proceedings of IDSCS 2020, pp. 201-209. https://doi.org/10.1007/978-981-15-5309-7_21

[17] Joseph, S., George, J. P. (2021). Offline character recognition of handwritten MODI script using wavelet transform and decision tree classifier. In Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces, pp. 509-517. https://doi.org/10.1007/978-981-16-0739-4_48

[18] Rajan, V. (2018). Aksharamukha: Script Converter. https://aksharamukha.appspot.com/converter.

[19] Sawant, S., Sharma, A., Suvarna, G., Tanna, T., Kulkarni, S. (2020). Word transcription of MODI script to Devanagari using deep neural network. In 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA), Mumbai, India, pp. 18-22. https://doi.org/10.1109/CSCITA47329.2020.9137781

[20] TDIL. (2015). Marathi General Text Corpus. Ministry of Electronics & Information Technology. Government of India. https://www.meity.gov.in/content/technology-development-indian-languages-tdil

[21] IIT Bombay. (2010). Marathidata. CFILT, IIT Bombay. https://www.cfilt.iitb.ac.in/.

[22] MihailJP. (2019). MarathiCursive: Unicode Modi script font. https://github.com/MihailJP/MarathiCursive, https://github.com/MihailJP/MarathiCursive/releases

[23] OCRopy. (2016). Open source document analysis and OCR system. GitHub. Inc. https://github.com/ocropus/ocropy.

[24] Rice, S.V., Nartker, T.A. (1996). The ISRI analytic tools for OCR evaluation. UNLV/Information Science Research Institute, TR-96, 2.

[25] Unicode. (2022). Modi Unicode. http://www.unicode.org/charts/PDF/U11600.pdf.

[26] Yalniz, I.Z., Manmatha, R. (2011). A fast alignment scheme for automatic OCR evaluation of books. In 2011 International Conference on Document Analysis and Recognition, pp. 754-758. https://doi.org/10.1109/ICDAR.2011.157