Optical Character Recognition Of Sanskrit Manuscripts Using Convolution Neural Networks

Article · January 2021

2 authors:

Bhavesh C Kataria
Kadi Sarva Vishwavidyalaya

Harikrishna B Jethva
Gujarat Technological University

Optical Character Recognition Of Sanskrit Manuscripts Using Convolution Neural Networks

Bhavesh Kataria[1*], Dr. Harikrishna B. Jethva[2]

[1]Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India

[2]Associate Professor, Department of Computer Engineering, Government Engineering College, Patan, Gujarat, India

---

ABSTRACT

Sanskrit is a 3,500-year-old Indian language and the liturgical language of Hinduism, Buddhism, and Jainism. Due to resemblance in the forms of distinct letters, script complication, non-forte in the representation, and a large number of symbols, the current study on Sanskrit Character Recognition from images of text documents is one of the most challenging. The Sanskrit language is written in the Devanagari script. There are a variety of approaches for recognizing characters in a scanned image [1,2,3,4,5].

This research provides an optical character recognition (OCR) system that enables to analyse the word recognition and translate various types of Sanskrit documents or images into text using deep learning architectures which include Recurrent Neural Networks (RNN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BLSTM) networks.

Existing methods focus only upon the single touching characters. But we also focus on designing a robust architecture for overlapping lines, touching characters in the middle and upper zone and half character which would increase the accuracy of the present OCR system for recognition of Sanskrit literature.

The results of the proposed system yield good recognition accuracy rates comparable to that of other character recognition systems.

Keywords : OCR, LSTM, BLSTM, SVM, ANN, Hidden Markov Model

I. INTRODUCTION

Most of the greatest literature works to come out of India were written in Sanskrit. It is considered the mother tongue of all contemporary languages. The Vedas, which were written in sanskrit, represent the spirit of Indian culture and history. The core thoughts of Buddhism were also written in sanskrit.

In many academic organizations, the existence of historical scientific and mathematical research work published in Sanskrit.

Scientists from all across the world are working on decoding ancient documents. However, one major stumbling block is the lack of adequately scanned and classified Sanskrit Texts. Furthermore, the issue is aggravated by poor maintenance and text quality. As a result, digitising historical documents that are not only significant for study but also represent an important part of India's culture and history.
With the emergence of digital content, the need for the development of an Optical Character Recognition (OCR) System with high performance has become essential. OCR research is a field of Pattern Recognition. Its process of human reading is the driving force behind the development of a machine that can read characters as well as humans.

OCR is mostly used for online and offline character recognition. Machine printed and handwritten characters are two types of offline character recognition. In machine printed documents have overlapping lines, touching characters in the middle and upper zone and half character [1]. These challenges bring researchers together to solve difficulties in recognition.

Since the last 20 years, as the need for devanagari documents has grown, numerous printed and handwritten monolingual character identification algorithms have emerged. Several OCRs for Indian languages such as Hindi, Bangla, and Telugu have been created [5,6,7,8,9,10,12,13]. However, there has been little or no effort to produce acceptable OCRs for Sanskrit.

Likewise, there is a need to address the issue of OCR of Sanskrit documents. Basic block diagram of OCR system is shown in below figure 1. Pre-processing, segmentation, representation, training and recognition, and postprocessing are the five primary processes.

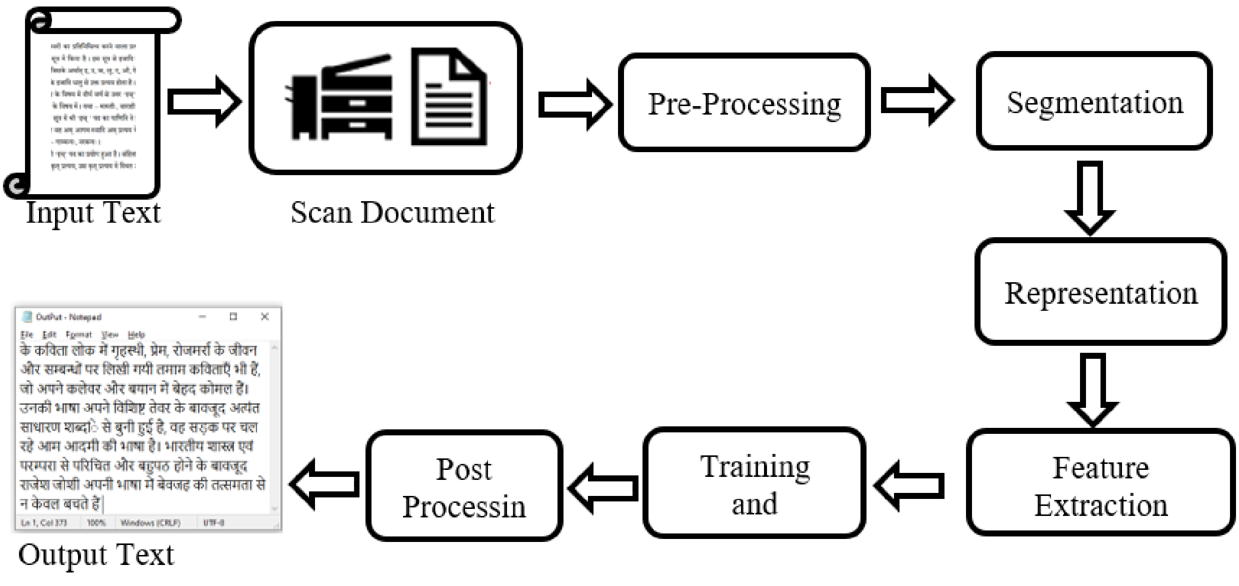


Figure 1. The components of an OCR system

Smoothing, sharpening, binarizing the image, removing the background, and extracting the needed information all require pre-processing to make the raw data useful in the descriptive stages of character analysis. In segmentation a task that breaks down an image of an arrangement of characters into sub-images of individual images by first segmenting the lines, then line segments the words, and finally from words to individual character. The yield of the segmentation phase is confined characters of the content [22]. The diverse strategies for feature extraction are actualized on the segregated character to get the feature vectors. These capabilities are utilized for the classification issue. At the representation step, a set of features is extracted to distinguish one image class from another. Then the classifiers that are used for training and recognition are k-NN, Linear-SVM, Neural Network, Deep Learning (LSTM) [11, 29,31].

II. RELATED WORKS

In 1973, segmentation-based algorithms were used to recognise Devanagari characters in printed texts for the first time. Many more studies have been conducted on the recognition of manuscripts in various Indian languages; most classical OCRs employ character segmentation to extract symbols and then recognise them using a classifier.

A neural network-based pattern recognition has attracted a lot of attention. The advancement of technology in terms of scanning devices, computation power and new learning strategies, Large neural networks can be developed to better and better approximate a good mapping. Unquestionably the invention of deep learning (particularly convolutional neural networks) is a paradigm that has already had a significant impact [26, 27, 28].

In current research, machine learning methods such as support vector machines (SVMs) and artificial neural networks (ANNs) [29, 31] are used to categorise characters in images in an OCR system for Indian languages. Existing Indic OCRs perform poorly on deteriorated or badly kept documents or materials, and their digitization capabilities are confined to high-quality text documents [41].

Using LSTM Neural Networks, Kundaikar T., et al. [40] worked on Multi-font Devanagari Text Recognition. In this paper, they used OCR on a scanned document with text in Devanagari script in multiple font styles, notably Nakula, Baloo, Dekko, Biryani, and Aparajita. In 2015, research adopted a segmentation-free strategy for word classification in printed Devanagari documents [39].

Rajib Ghosh et al. proposed employing horizontal zoning for RNN-based online handwritten word recognition in Devanagari and Bengali scripts. They employed two recently developed Recurrent Neural Network (RNN) models, LSTM and BLSTM, to recognize online handwritten cursive and non-cursive words in Devanagari and Bengali scripts. [32]

We have effectively implemented Optical Character recognition using recurrent neutral networks, which are a type of deep learning with long-term memory cell [26, 27, 28, 38]. Recent developments in the deep BLSTM network have increased interest in text recognition, obtained in excellent results.

[35,36,37], for multiple layered networks, this approach has been success fully applied for Sanskrit Manuscripts.

Table 1. Observation : All experimental methods had a recognition rate (2000-2020)

No	Method	Classifier	Recog. Rate %
1	Histogram Projection (Binary)	SVM	26.313
2	Celled Projection (Binary)	SVM	49.9414
3	Celled Projection (Binary)	k-NN	76.1632
4	Distance Profile (Binary)	SVM	40.1277
5	Distance Profile (Binary)	k-NN	58.947
6	Distance Profile (Skeleton)	SVM	36.7653
7	Crossing (Binary)	SVM	15.0007
8	Zoning (Binary)	SVM	50.6451
9	Zoning (Binary)	k-NN	78.5351
10	Zoning (Skeleton)	SVM	41.848
11	Zoning (Grayscale)	SVM	52.4176
12	Zoning (Grayscale)	k-NN	66.128
13	Gradient Feature (Gray)	SVM	60.0417
14	Gradient Feature (Gray)	k-NN	72.5792
15	Moment Hu (Gray)	SVM	33.481
16	Moment Hu (Gray)	k-NN	33.481
17	HoG (Gray)	SVM	71.2759
18	HoG (Gray)	k-NN	84.3477
19	NPW (Binary)	SVM	51.388
20	NPW (Gray)	SVM	54.1249
21	Kirsch (Gray)	SVM	62.4528
22	HoG with Zoning (Gray)	SVM	69.6859
23	HoG with Zoning (Gray)	k-NN	83.5006
24	NPW-Kirsch (Gray)	SVM	63.5736
25	NPW-Kirsch (Gray)	k-NN	76.7105
26	HoG on Kirsch edge (Gray)	k-NN	82.0931
27	HoG + NPW-Kirsch (Gray)	k-NN	84.7517
28	Zoning + Celled Projection (Binary)	k-NN	77.701
29	HoG + NPW-Kirsch (Gray) + Zoning (Binary)	k-NN	85.1557
30	Convolutional Neural Network	ANN	90.3086
31	Recurrent Neural Networks	LSTM-BLSTM)	93.15 In Process

2.1 Problem Definition

In current digital age, the digitization of documents become obligatory to have all the available information in a digital form recognized by machines, A direct solution to the use of character recognition systems to convert document images into text, this requires research in the area optical character recognition (OCR).

The following conclusions can be drawn based on the extensive study of various algorithms and Methods on OCR

- ✓ In may literature most of algorithms and study have directly paid attention on the major issues related with OCR of English and other regional manuscripts.
- ✓ Most have attempted to employ segmentation, edge detection, and classifiers for training and recognition such as ANN, SVM, Neural Network, and CNN of characters from printed and handwritten documents.
- ✓ The classification and analysis of complex/conjoin characters is important in character recognition, that has not addressed clearly for Sanskrit Manuscripts. So, there is a strong need of quick and accurate character recognition for Sanskrit Manuscripts.
- ✓ The is need of ground-truthed database sanskrit character recognition systems

III. OBJECTIVE AND SCOPE OF WORK

3.1 Objective

Recognition of textual content in document images. To this end, an attempt is made to:

- ✓ To make document analysis and comprehension easier, examine Sanskrit Manuscripts and their language norms.
- ✓ Create an OCR for recognising printed document images in Sanskrit, as language that has received less attention in the past.
- ✓ Propose a self-adaptable OCR architecture and learning algorithms that can learn and adapt to new pages or documents to increase performance.
- ✓ Performance analysis of OCRs on low-quality, variable-printing synthetic and real-life document images collections.
- ✓ Design feature extraction, classification, and feedback mechanisms for document image recognition using machine learning algorithms.
- ✓ The features such as binary codes are extracted from the characters. The neural network classifier is built using Bidirectional Long Short-Term Memory (BLSTM) network which is trained using a Sanskrit character dataset. The neural network (CNN, RNN, BLSTM) is used to test the input images. The output is provided as a text document with the recognized words. Since the input feed is obtained from images, the noise will be high compared to the existing system input set which uses scanned images. Noise reduction technique such as low intensity pixel removal is applied to reduce the noise from the input image for improving the efficiency.

3.2 Scope of work

The goal of this research is to advance the state-of-the-art in document image recognition.
The images of printed documents that are archived in digital libraries and other apps are our main emphasis here.

We use machine learning methodologies in proposing algorithms for feature extraction and classification, as well as in designing feedback mechanism for document images recognition. Here Bidirectional Long Short-Term Memory (LSTM) is a type of deep learning artificial recurrent neural network (RNN). To benchmark the performance of the BLSTM networks, they are initially tested on typical Sanskrit datasets. They outperform any other modern OCR technology in terms of recognition without the use of advanced features or language modelling. As a result, their use has been expanded to include more complex scripts such as Sanskrit and Devanagari.

IV. METHODS AND MATERIAL

4.1 MODEL DESCRIPTION - NEURAL NETWORKS

RNN : Recurrent neural networks (RNNs) can model sequence data using their recurrent connections, which map all previous inputs to each output and construct a memory. Bidirectional RNNs help the network learn context both forward and backward.

Rectified Linear Unit

- The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do is that, images are naturally non-linear (do all negative values to 0 in the feature map).
- When you look at any image, you'll find it contains a lot of non-linear features (e.g. the transition between pixels, the borders, the colors, etc.).
- The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation.

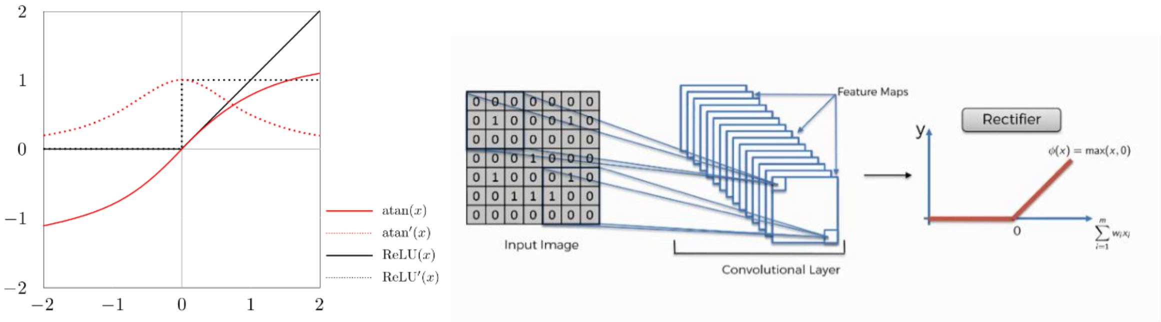


Figure 2. A plot of the atan and ReLU activation function and their derivative

As seen in Fig. 2, the Rectified linear function is essentially a hinge function that is zero for negative input values and the identical function otherwise. The function is incredibly fast to compute and has a simple derivative, with a value of 0 for negative input values and a value of 1 otherwise. Typically, the function is trimmed so that it does not exceed a significant value, such as 20. The absence of differentiability when x = 0 isn't a big deal because it may be adjusted to 0 or 1 arbitrarily. Xavier Glorot et al [28] suggest that this basic activation function can be utilised to train convolutional neural networks (deep learning) effectively without any unsupervised pre-training. Whereas layer-wise, unsupervised pre-training or ReLU units help to deal with many hidden layers, the vanishing gradient problem for recurrent neural networks can be successfully avoided with an LSTM architecture [17].

LSTM

An LSTM network is a recurrent neural network that has LSTM cell blocks in place of our standard neural network layers. These cells have various components called the input gate, the forget gate, and the output gate shown as below.

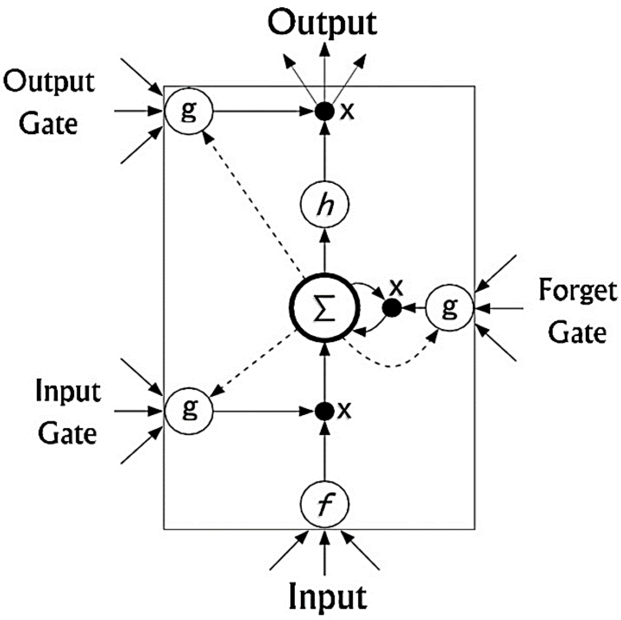


Figure 3. LSTM memory Cell - Gate functions are labeled as g, input activation as f, output activation as h, multiplication nodes as x and the core node, which realizes a summation, as ∑.

A LSTM unit is a second-order recurrent node, which means that the weight of the recurrent connection, the connections toward the node, and the connections leaving the node are not fixed and are determined by the activation of dedicated gate nodes. Such a LSTM node is depicted in Fig. 3. The core of the LSTM node, ∑, is a summation node with a recurrent connection of weight 1. The input from the network after being squashed by node f, the recurrent connection, as well as the squashed core value are each multiplied by the activation of the gate functions g, usually a logistic function between 0 and 1. This turns an LSTM cell basically into a memory cell, not unlike a computer memory, which can be set or reset, and is left unchanged otherwise.

BLSTM :

Bidirectional LSTM [43] is optimized LSTM, which can read input sequences from both ends. This structure enables LSTM to learn sequential patterns from both directions. When transcribing letters, words, or word sequences, recurrent neural networks (RNNs) can access a wide range of context [44,45]. Bidirectional RNNs (BRNNs) can incorporate context on both sides of every point in the input sequence, whereas normal RNNs only use previous context. This is useful in handwriting identification since identifying a letter typically necessitates looking at both the context to the right and left of it.

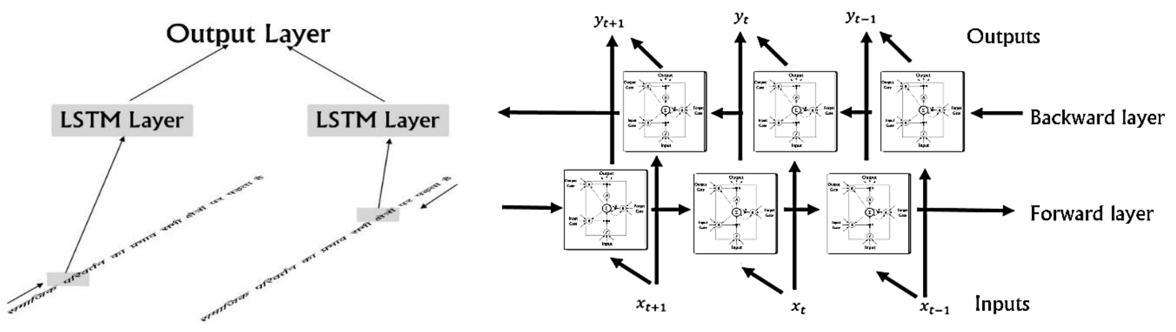


Figure 4. Bidirectional Recurrent Neural Network Layer

4.2 PROPOSED MODEL

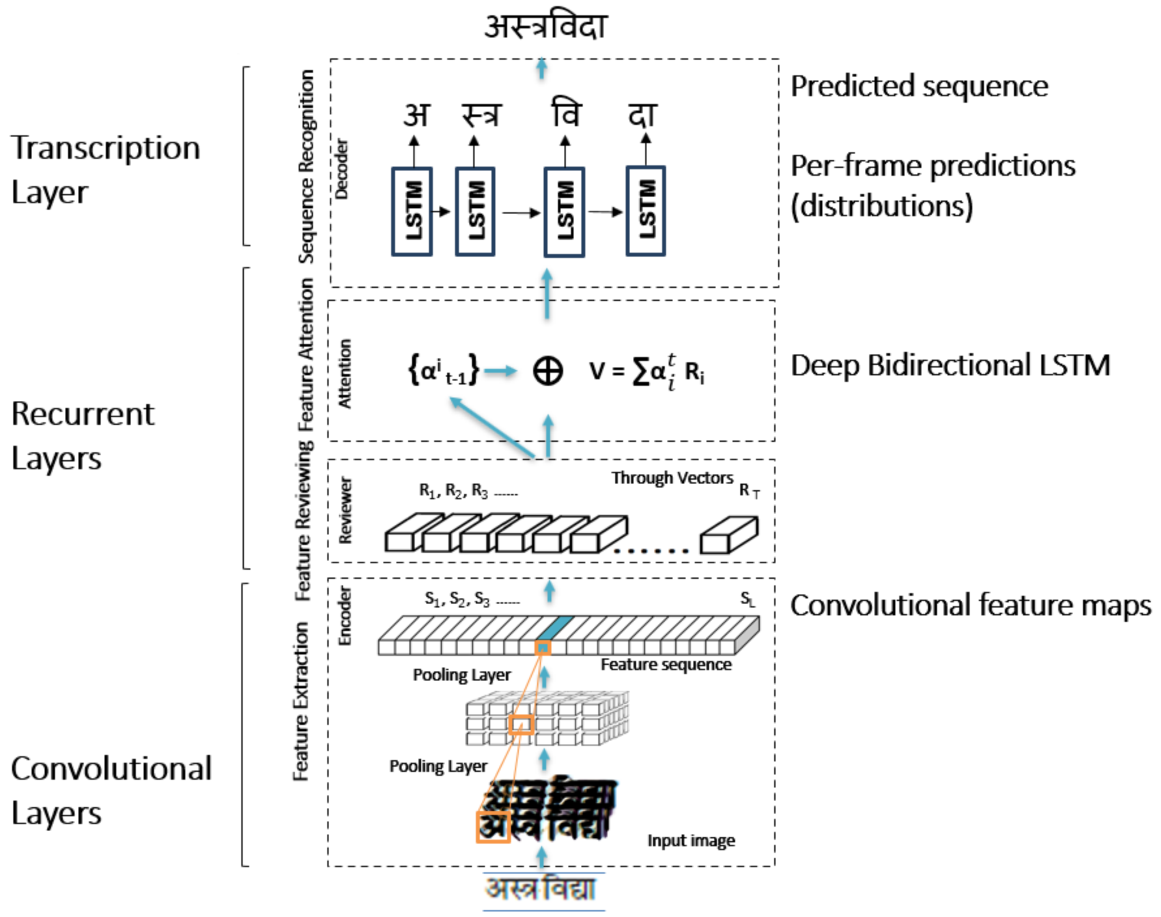


Figure 5 : The network architecture.

From bottom to top, the Proposed network architecture comprises of three components: convolutional layers, recurrent layers, and a transcription layer, as shown in Fig. 5.

1) Convolutional layers, from the input picture, which extracts a feature sequence;
2) Recurrent layers, which forecasts each frame's label distribution;
3) Transcription layer, which converts per-frame predictions into a final label sequence

1) Feature Sequence Extraction

The convolutional layer component of the model is built by combining the convolutional and max-pooling layers from a typical CNN model (fully-connected layers are removed). A sequential feature representation is extracted from an input picture using this component. All photos must be resized to the same height before being supplied into the network. The input for the recurrent layers is therefore a sequence of feature vectors derived from the feature maps produced by the components of convolutional layers. Each feature vector in a feature sequence is formed on the feature maps by column, from left to right. This indicates that the i-th feature vector is the concatenation of all the maps' i-th columns. In our settings, the width of each column is set to a single pixel.

2) Sequence Labeling

As the recurrent layers, a deep bidirectional Recurrent Neural Network is formed on top of the convolutional layers. For each frame xt in the feature sequence $S = S_1, S_2. S_3.., S_L$, the recurrent layers predict a label distribution yt [Fig 4,5]. The recurrent layers have three distinct benefits. To begin with, RNN excels at capturing contextual information inside a sequence. For image-based sequence recognition, using contextual clues is more reliable and useful than considering each sign separately. In the case of scene text recognition, broad characters may necessitate numerous frames to completely represent.

3) Transcription layer

The process of translating RNN's per-frame predictions into a label sequence is known as transcription. The goal of transcription, mathematically, is to discover the label sequence with the best likelihood based on per-frame predictions.

4) Network Training

a). SANSKRIT CHARACTERS ARE USED FOR RESEARCH WORK

Devanagari Alphabet

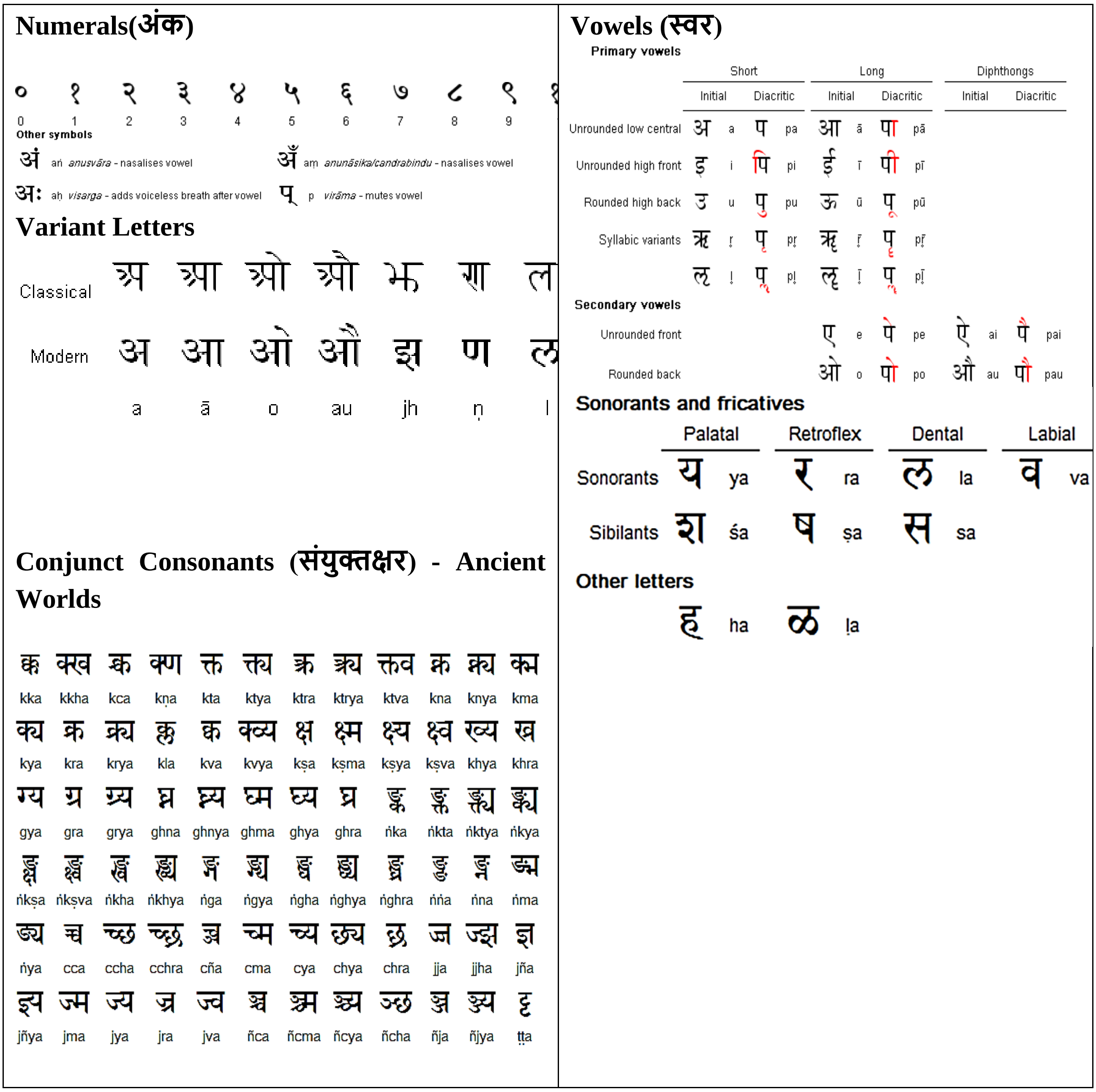


Figure 6. The Language written with the Devanagari alphabet

As show in figure 6 Sanskrit text contains of several compound characters which are formed by different combinations of half letter and full letter consonants. Some examples of compound characters are shown in Fig 6. Since such compound characters are either less frequent or completely absent in Hindi text, Hindi OCRs would not be trained to segment and classify such characters correctly. Subsequently, the Hindi OCRs would display poor results in Sanskrit text.

b). DATASETS

For training and benchmarking, many character recognition algorithms require a large amount of ground-truthed real-world data. A trainable OCR model's generalisation accuracy is directly affected by the quantity and quality of training data.

In our proposed datasets we have taken data from two sources
- ✓ Existing Sanskrit document image texts and synthetically generated Sanskrit texts
- ✓ **Scan image Characters -** Annotated 19,579 lines of Sanskrit text from four different books.

Books
- महात्मा गांधी का आर्थिक एवं सामाजिक दर्शन- Economic and Social Philosophy of Mahatma Gandhi
- Bhrigu Sanghita By Maharshi Bhrigu
- श्रीमदभगवद्गीता - Bhagavad Gita - The Song of God By Swami Mukundananda
- मनमनाभव (श्रीमद्भागवतगीता पर आधारित)- Manmanabhav (Based on Shrimad Bhagwat Geeta)

To ensure high data quality, the notations were acquired from Sanskrit domain specialists. The relevant statistics are listed in Table 2.

Table 2. Statistics of our annotated datasets

Book	Pages	Lines	Words
Economic and Social Philosophy	152	5490	38320
Bhrigu Sanghita	130	4696	32774
Bhagavad Gita	110	3973	27731
Manmanabhav	150	5418	37816
Total	542	19579	136642

Sanskrit writings synthesised - typefaces from many sources [22,23,24,25]. The list was then reduced to 67 fonts that allow conjunct character rendering. We used 4500 different lines per typeface from old Sanskrit literature. Available at https://sanskritdocuments.org to increase the diversity of synthetic data. [Section A Font Styles]

c). DATA PREPARATION

For our synthetic data, we employ binary images and trim the output text to remove superfluous whi tespace around the actual text content. There are 170 characters in the vocabulary set. All of the photos have been scaled to 32 pixels in height, with the width altered to match the original aspect ratio.

d). OPTIMIZATION

For optimization, we use the ADADELTA (with decay rate of 0.95) [42] to automatically calculate per-dimension learning rates. There is no need to manually establish a learning rate with ADADELTA. More notably, we discovered that ADADELTA optimization converges faster than the momentum technique. The learning rate is initially set at 1. We train with 16-bit mini-batches and stop after 310k iterations.

e). Training

$X = \{S_i, R_i\}_i$ denotes the training dataset, with Si representing the training picture and Ri representing the ground truth label sequence. The goal is to reduce the negative log-likelihood of the conditional probability of ground truth to the minimum possible value:

$$O = -\sum_{S_i, R_i \in X} \log p(R_i \mid Y_i)$$

where Y_i denotes the sequence generated by S_i's recurrent and convolutional layers. This objective function immediately derives a cost value from an image and the ground truth label sequence. As a result, the network may be trained end-to-end on pairs of pictures and sequences, obviating the need to identify all individual components in training images manually.

V. RESULTS AND COMPARISONS

We tested the proposed BLSTM based CRNN model's efficacy using recognised benchmarks for scene text recognition.

Table 4 : Network Configuration details

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2 × 2, s:1, p:0
Max Pooling	Window:1 × 2, s:2
Batch Normalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
Batch Normalization	-
Convolution	#maps:512, k:3 × 3, s:1, p:1
Max Pooling	Window:1 × 2, s:2
Convolution	#maps:256, k:3 × 3, s:1, p:1
Convolution	#maps:256, k:3 × 3, s:1, p:1

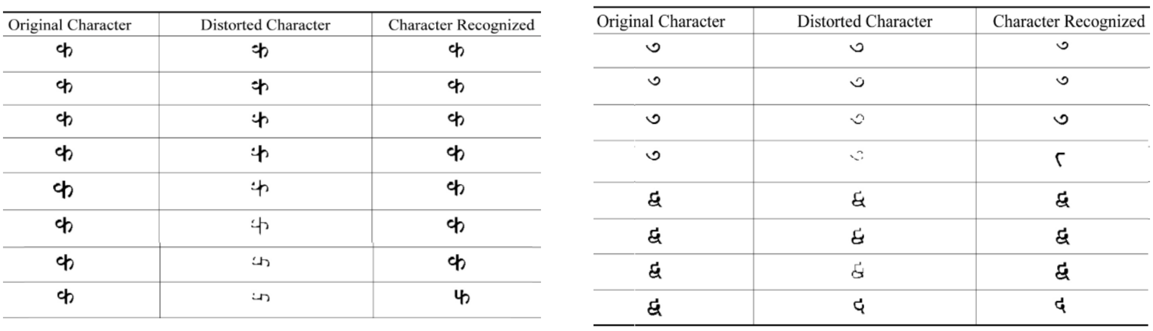
Max Pooling	Window:2 × 2, s:2
Convolution	#maps:128, k:3 × 3, s:1, p:1
Max Pooling	Window:2 × 2, s:2
Convolution	#maps:64, k:3 × 3, s:1, p:1
Input	W × 32 gray-scale image

The input images are expanded to a fixed width of 512 and sent to a 7-layer CNN network to obtain line image characteristics. The CNN features are fed into the encoder RNN, which is a 256-hidden-unit BLSTM model. The decoder consists of two layers of LSTMs, each with 128 hidden layers. (Seen at the top of Figure 5), and eqs. 1 to 6 were applied.

Robustness of Classifier

The performance of the classifier at alphabet level is shown in Table 5. To test the robustness of the classifier we gradually increased the distortion in the input character images and observed the recognized characters thereof. Table 5 shows the original character, the distorted character, provided as an input to the classifier, and the output character recognized by the classifier. By observing the table, we can say that the classifier is robust enough and is able to correctly classify even highly distorted characters too. The only instances of misclassifications are at the level when even a human expert would guess incorrectly. For example, for the Hindi letter ‘ka’ continuous distortion was incorporated, and the classification results observed thereafter are shown from rows number 1 to 8 of Table 5. Until row number 7 the classifier was perfectly able to recognize it as ‘ka’, but it failed at row number 8, recognizing it as ‘ph’. It is almost certain that even if human experts are provided with the same distorted letter, as shown in row number 9 under the column ‘Distorted Character’ then most of the time they will recognize it as ‘ph’ too. Similar results were obtained for many other letters too such as ‘chha’ and ‘dwa’, as shown at rows number 14 and 27 respectively of Table 5. So, this all proves the robustness of the classifier developed using the BLSTM approach.

Table 5. Robustness of the classifier



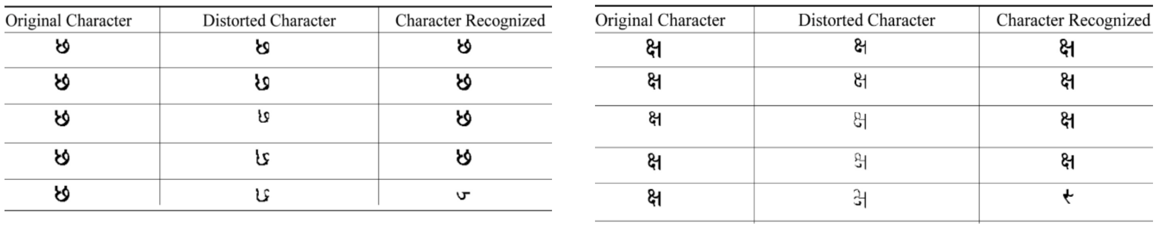


Table 6. Result comparison of OCR Systems for Printed Devanagari Sanskrit Characters

Method	Feature	Classifier	Data Set Size	Accuracy (%)
Govindraju et al. [18]	Gradient	Neural Networks	4,506	84
Kompalli et al. [19]	GSC	Neural Network	32,413	84.77
Bansal et al. [16]	Statistical Structural	Statistical Knowledge Sources	Unspecified	87
Huanfeng Ma et al. [20]	Statistical Structural	Hausd or off image comparison	2,727	88.24
Natrajan et al. [25]	Derivatives	HMM	21,727	88.24
Bansal et al. [17]	Filters	Five filters	Unspecified	93
Dhurandhar et al. [23]	Contours	Interpolation	546	93.03
Kompalli et al. [22]	GSC	K-nearest neighbor	9,297	95
Kompalli et al. [21]	SFSA	Stochastic finite state automa- ton	10,606	96
Dhingra et al. [24]	Gabor	MCE	30,000	98.5
Bhavesh et al.	Gradient	Stochastic Gradient Descent (SGD)	34,215	98.64

Table 7. Result comparison of OCR Systems for Printed Devanagari words

Method	Feature	Classifier	Data Set Size	Accura cy (%)
Govindraju et al. [18]	Gradient	Neural Networks	4,506	53
Kompalli et al. [22]	GSC	K-nearest neighbor	1,882	58.51
Kompalli et al. [19]	GSC	Neural Network	14,353	61.8

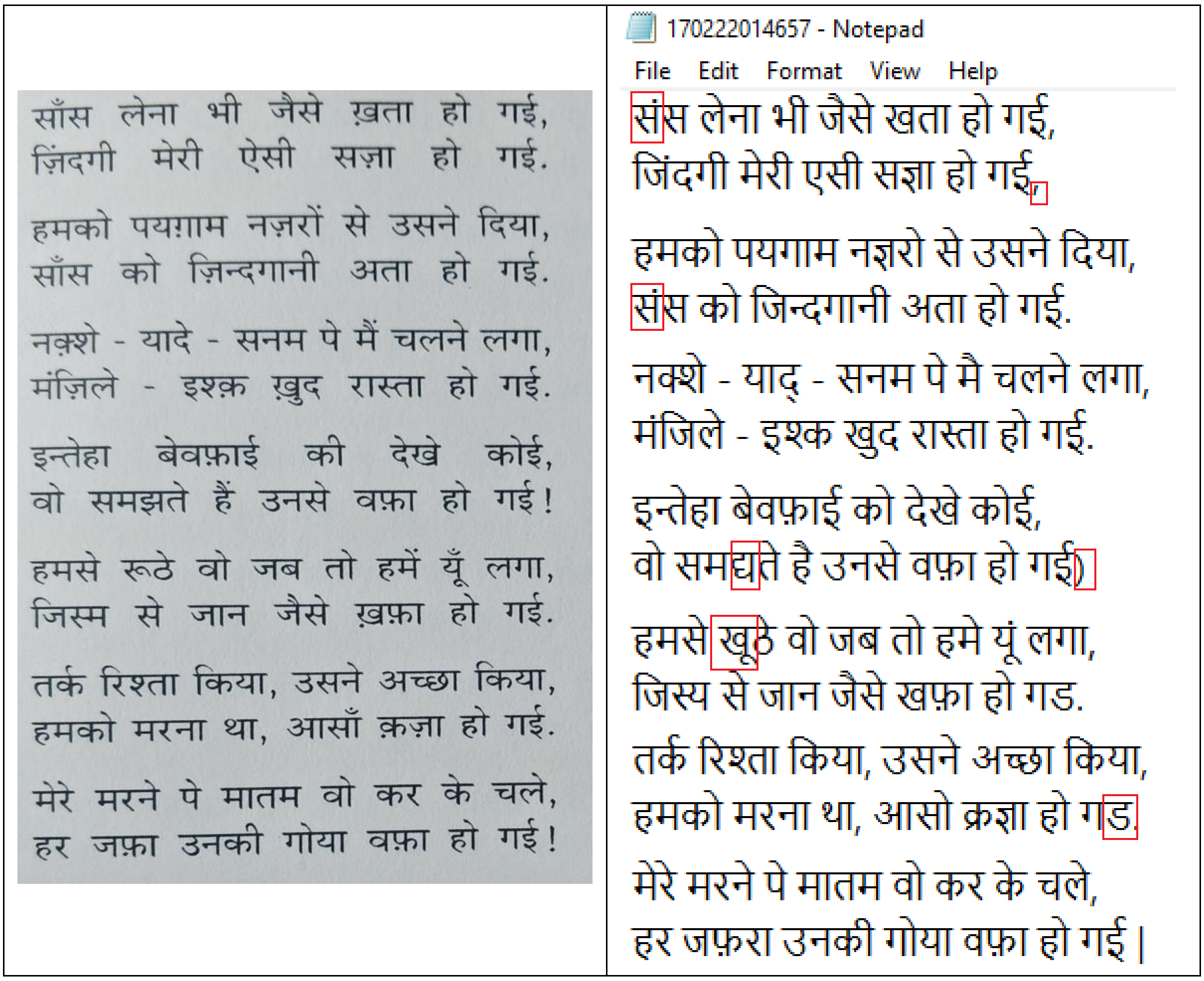
Ma et al. [20]	Statistical Structural	Hausdorff image comparison	2,727	66.78
Kompalli et al. [21]	SFSA	Stochastic finite state automaton	10,606	87
Bhavesh et al.	Gradient	Stochastic Gradient Descent (SGD)	15,455	97.3

We tested our classifier on various printed Sanskrit documents and gathered the results. In fact the testing was done at two levels: individual letter level and paragraph level. The character level testing was performed on approximately 34,215 individual characters. Paragraph level testing was performed on approx. 144 paragraphs of different Sanskrit fonts consisting of approximately 15,455 words (including devanagari characters, numeric digits, and punctuation symbols). The character level performance was excellent with a correct recognition rate of 98.6%. However, at paragraph level the performance dropped and an average accuracy of approximately 93.54% was achieved and is shown in Table 6 and Table 7 respectively.

Two such input sample paragraph used for testing the performance of the classifier is shown in Table 8. It can be observed that the recognition rate is higher for individual than for continuous characters

Table 8. Testing performance of the classifier

Input Image	Output Text
(७) तन्नियोगे नियुक्तेन कृतं कृत्यं हनूमता । न चात्मा लघुतां नीतः सुग्रीवश्चापि तोषितः ।। (८) अहं च रघुवंशश्च लक्ष्मणश्च महाबलः । वैदेह्या दर्शनेनाद्य धर्मतः परिरक्षिता ।। (९) इदं तु मम दीनस्य मनो भूयः प्रकर्षति । यदि हास्य प्रियाख्यातुर्न कुर्मि सदृशं प्रियम् ।। (१०) एष सर्वस्वभूतस्तु परिष्वङ्गो हनूमतः । मया कालमिमं प्राप्य दत्तश्चास्तु महात्मनः ।।	*110222125952 - Notepad File Edit Format View Help (७) तन्नियोगे नियुक्तेन कतं कृत्यं हनूमता । न चात्मा लघुतां नोतः सुग्रीवश्चापि तोषितः ॥! (८) अहं च रघुवंशश्च लक्ष्मणश्च महाबलः । वेदेह्या दशनेनाद्य धर्मतः परिरक्षिता ॥ (९) इदं तु मम दीनस्य मनो भूयः प्रकषंति । यदि हास्य प्रियाख्यातुनं कुमि सदृशं प्रियम् ।। (१०) एष सर्वस्वभूतस्तु परिष्वडगो हनूमतः । मया कालमिमं प्राप्य दत्तश्चास्तु महात्मनः ।।



OCR Model Comparison

Table 9. Performance of OCR's character accuracy for text in Sanskrit Devanagari Font Style

OCR	Accuracy (%)		
	Image 1	Image 2	Image 3
Tesseract	82.15	93.14	92.14
Indsenz	73.54	85.65	86.65
E Aksharayan	71.61	79.21	81.21
Alex Net	80.64	84.60	86.60
Google Le Net	81.23	86.54	87.54
ResNet-50	86.15	91.24	89.24
SVM	79.40	82.40	85.40
Bidirectional LSTM	94.56	96.63	98.64

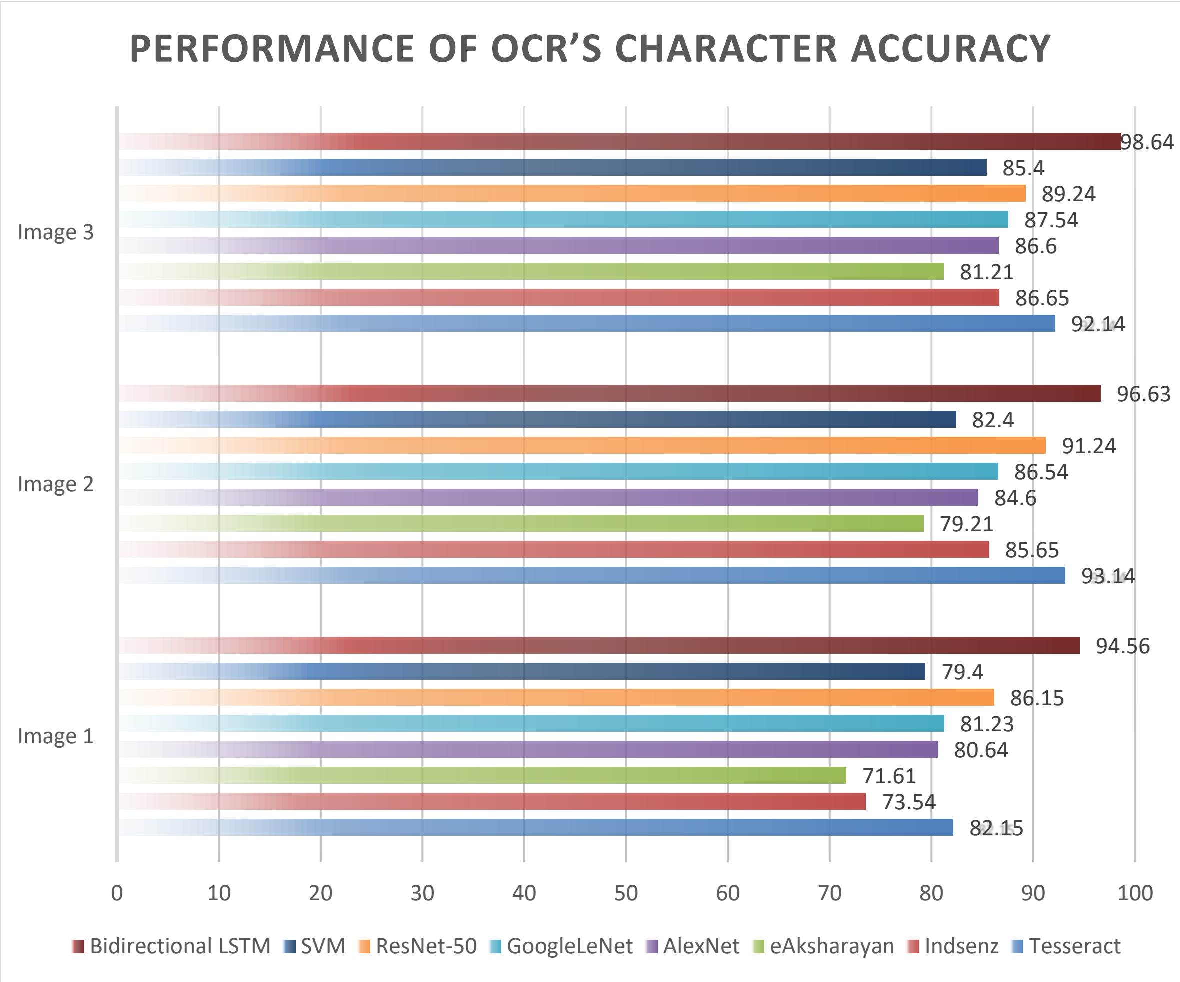
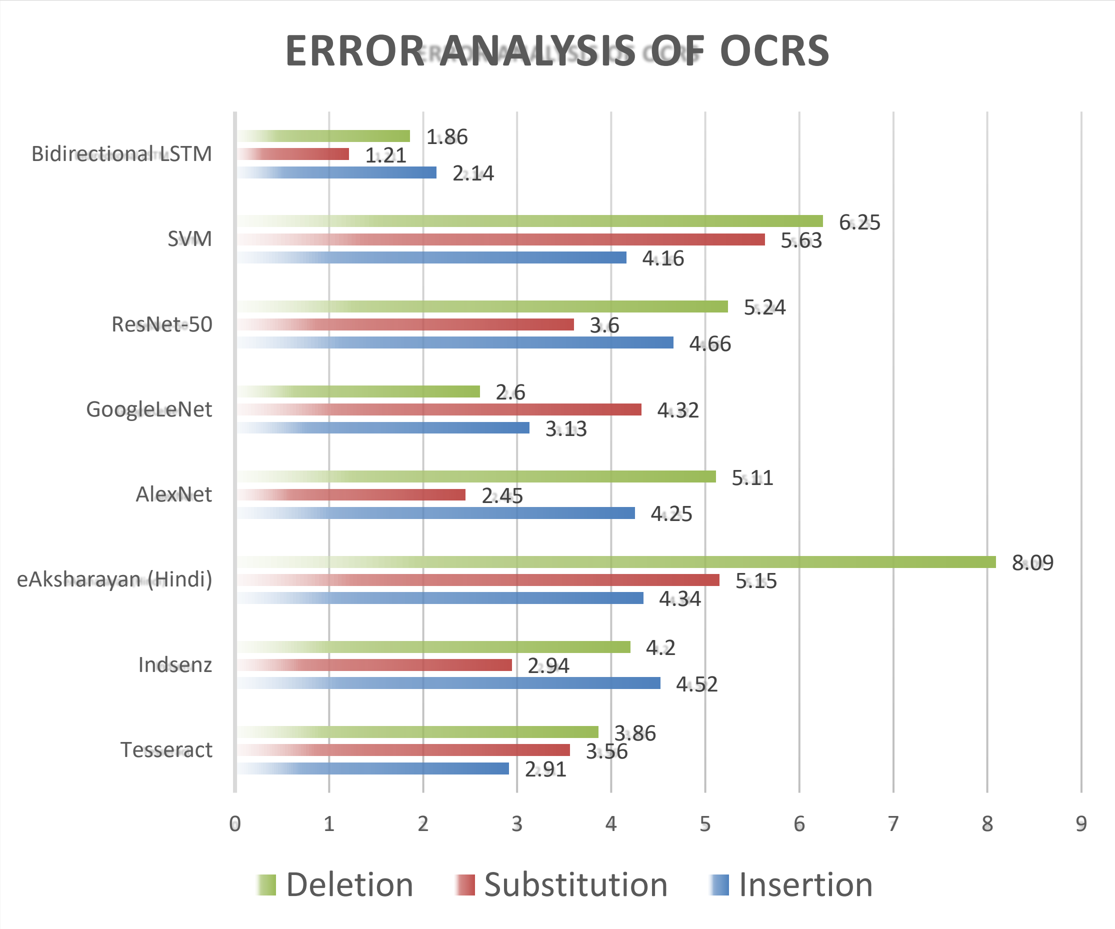


Table 10. Error analysis of OCRs for text document image in Sanskrit Devanagari Font Style

OCR	Type of error		
	Insertion	Substitution	Deletion
Tesseract	2.91	3.56	3.86
Indsenz	4.52	2.94	4.20
E Aksharayan (Hindi)	4.34	5.15	8.09
Alex Net	4.25	2.45	5.11
Google Le Net	3.13	4.32	2.60
ResNet-50	4.66	3.60	5.24
SVM	4.16	5.63	6.25
Bidirectional LSTM	2.14	1.21	1.86



VI. CONCLUSION

Presently, the work is being done on using Convolutional Neural Networks to recognise Sanskrit (Devanagari) characters (LSTM and BLSTM). An experiment was conducted on a large dataset to test the performance of new LSTM-BSLTM based Convolutional Neural Network techniques for sanskrit character recognition. The study lays the path for the creation of high-performance OCRs that can be used to the huge traditional Indian document collections that are currently available. We want to improve the LSTM-BLSTM model in the future, with a goal of lowering WER even further, incorporating OCR correction systems [35], and reducing the dependency on real data for training.

VII. REFERENCES

[1]. D. Yadav, S. Sanchez-Cuadrado, and J. Morato, "Optical Character Recognition for Hindi Language Using a Neural-network Approach" Journal of Information Processing Systems, vol. 9, no. 1, pp. 117–140, Mar. 2013.

[2]. Agravatt, Shweta & Goswami, Mukesh & Agravatt, Hardik. (2015). A New Approach for Zone Identification on Printed Gujarati Text: Vertical Bar Method. International Journal of Software & Hardware Research in Engineering. 3. 89-93.

[3]. K. R.Dahake, S. R. Suralkar, and S. P. Ramteke, "Optical Character Recognition for Marathi Text Newsprint," International Journal of Computer Applications, vol. 62, no. 16, pp. 11–15, Jan. 2013.

[4]. P. M. Kamble and R. S. Hegadi, "Handwritten Marathi Character Recognition Using R-HOG Feature," Procedia Computer Science, vol. 45, pp. 266–274, 2015.

[5]. Bhavesh Kataria, Dr. Harikrishna B. Jethva. (2020). Sanskrit Character Recognition using Convolutional Neural Networks: A Survey. International Journal of Advanced Science and Technology, 29(7), 1059 - 1071.

[6]. S. Aghav and S. S. Paygude, "Computer Assisted Printed Character Recognition in Document Based Images," Procedia Engineering, vol. 38, pp. 3222–3227, 2012.

[7]. Ahmed Mahdi Obaid, Hazem M. El Bakry, M.A. Eldosuky, A.I. Shehab, Handwritten Text Recognition System Based on Neural Network, International Journal of Advanced Research inComputer Science & Technology, Vol. 4, Issue 1 (Jan. - Mar. 2016) pp 72-77

[8]. N. S. Panyam, V. L. T.R., R. Krishnan, and K. R. N.V., "Modeling of palm leaf character recognition system using transform based techniques," Pattern Recognition Letters, vol. 84, pp. 29–34, Dec. 2016.

[9]. K. S. Varghese, A. James, and S. Chandran, "A Novel Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition," Procedia Technology, vol. 24, pp. 1333–1340, 2016.

[10]. R. Smith, D. Antonova, and D.-S. Lee, "Adapting the Tesseract open-source OCR engine for multilingual OCR," in Proceedings of the International Workshop on Multilingual OCR - MOCR '09, 2009

[11]. V. A. Naik and A. A. Desai, "Online handwritten Gujarati character recognition using SVM, MLP, and K-NN," in 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017.

[12]. Xiaoqing Ding, Li Chen, and Tao Wu, "Character Independent Font Recognition on a Single Chinese Character," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 2, pp. 195–204, Feb. 2007

[13]. S. Bag, G. Harit, and P. Bhowmick, "Recognition of Bangla compound characters using structural decomposition," Pattern Recognition, vol. 47, no. 3, pp. 1187–1201, Mar. 2014.

[14]. Desai, "Gujarati handwritten numeral optical character reorganization through neural network," Pattern Recognition, vol. 43, no. 7, pp. 2582–2589, Jul. 2010.

[15]. N. Prameela, P. Anjusha, and R. Karthik, "Off-line Telugu handwritten characters recognition using optical character recognition," in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017.

[16]. Bansal, V., Sinha, R.M.K.: Integrating knowledge sources in Devanagari text recognition. IEEE Trans. Syst. Man Cybern. A: Syst. Hum., Vol.30, No.4, 2000, pp.500-505.

[17]. Bansal, V., Sinha, R.M.K.: A Complete OCR for Printed Hindi Text in Devanagari Script. Sixth In- ternational Conference on Document Analysis and Recognition, IEEE publication, Seatle USA., 2001, pp.800-804.

[18]. Govindaraju, V., Khedekar, S., Kompalli, S., Farooq, F., Setlur, S., Vemulapati, R.: Tools for enabling digital access to multilingual Indian documents. in Proc. 1st Int. Workshop Document Image Anal. Libraries, 2004, pp.122-133.
[19]. Kompalli, S., Nayak, S., Setlur, S., Govindaraju, V.: Challenges in OCR of Devanagari documents. In Proc. 8th Conf. Document Anal. Recognition, 2005, pp.1-5.
[20]. Ma, H., Doermann, D.: Adaptive Hindi OCR using generalized Hausdorff image comparison. ACM Trans. Asian Lang. Inf. Process., Vol.2, No.3, 2003, pp.193-218.
[21]. Kompalli, S., Setlur, S., Govindaraju, V.: Devanagari OCR using a recognition driven segmentation framework and stochastic language models. Int. J. Document Anal. Recognit., Vol.12, 2009, pp.123- 138.
[22]. Kompalli, S., Setlur, S., Govindaraju, V.: Design and comparison of segmentation driven and recog- nition driven Devanagari OCR. In Proceeding 2nd Int. Conf. Document Image Anal. Libraries, 2006, pp.1-7.
[23]. Dhurandhar, A., Shankarnarayanan, K., Jawale, R.: Robust pattern recognition scheme for Devanaga- ri script. Comput. Intell. Security, Part I, Lecture Notes in Artificial Intelligence 3801, 2005, pp.1021- 1026 .
[24]. Dhingra, K. D., Sanyal, S., Sanyal, P. K.: A robust OCR for degraded documents. In Lecture Notes in Electrical Engineering. Huang et al., Eds. New York: Springer-Verlag, 2008, pp.497-509.
[25]. Natarajan, P., MacRostie, E., Decerbo, M.: The BBN Byblos hindi OCR system. In Guide to OCR for Indic Scripts, V. Govindaraju and S. Setlur, Eds. New York: Springer-Verlag, 2009, pp.173-180.
[26]. Itamar Arel, Derek C. Rose, and Thomas P. Karnowski. Deep Machine Learning - A New Frontier in Artificial Intelligence Research. Computational Intelligence Magazine, IEEE, 5(4):13-18, 2010.
[27]. Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville. Deep Learning. Book in preparation for MIT Press, 2014.
[28]. Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. In 14th Int'l Conf. on Artificial Intelligence and Statistics, pages 315–323.
[29]. T. Thireou and M. Reczko. Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 4(3):441–446, 2007
[30]. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the Pastand the Future in Protein Secondary Structure Prediction. Bioinformatics, 15, 1999.
[31]. Chen and N. Chaudhari. Protein Secondary Structure Prediction with bidirectional LSTM networks. In International Joint Conference on Neural Networks: Post-Conference Workshop on Computational Intelligence Approaches for the Analysis of Bio-data (CI-BIO), August 2005.
[32]. R. Ghosh, A Recurrent Neural Network based deep learning model for offline signature verification and recognition system. Expert Systems With Applications (2020), doi: https://doi.org/10.1016/j.eswa.2020.114249.

[33]. A. L. Spitz. Multilingual Document Recognition. In H. Bunke and P. S. P. Wang, editors, Handbook of character Recognition and Document Image Analysis, pages 259–284. World Scientific Publishing Company, 1997.

[34]. Jurgen Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks, 61:85-117, 2015.

[35]. Paul, I. J. L., Sasirekha, S., Vishnu, D. R., & Surya, K. (2019). Recognition of handwritten text using long short term memory (LSTM) recurrent neural network (RNN). doi:10.1063/1.5097522

[36]. A. Dwivedi, R. Saluja and R. K. Sarvadevabhatla, "An OCR for Classical Indic Documents Containing Arbitrarily Long Words," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2386-2393, doi: 10.1109/CVPRW50498.2020.00288.

[37]. Dutta, K., Krishnan, P., Mathew, M., & Jawahar, C.V. (2018). Offline Handwriting Recognition on Devanagari Using a New Benchmark Dataset. 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), 25-30.

[38]. Bansal, Veena & Sinha, R.M.K.. (2000). Integrating knowledge sources in Devanagari text recognition system. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on. 30. 500 - 505. 10.1109/3468.852443.

[39]. T. Karayil, A. Ul-Hasan and T. M. Breuel, "A segmentation-free approach for printed Devanagari script recognition," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 946-950, doi: 10.1109/ICDAR.2015.7333901.

[40]. T. Kundaikar and J. D. Pawar, “Multi-font Devanagari Text Recognition Using LSTM Neural Networks,” in First International Conference on Sustainable Technologies for Computational Intelligence, 2020, pp. 495–506.

[41]. Holley, Rose. (2009). How good can it get? Analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. D-Lib Magazine: The Magazine of the Digital Library Forum. 15.

[42]. Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. Retrieved from http://arxiv.org/abs/1212.5701

[43]. A. Dwivedi, R. Saluja and R. K. Sarvadevabhatla, "An OCR for Classical Indic Documents Containing Arbitrarily Long Words," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2386-2393, doi: 10.1109/CVPRW50498.2020.00288.

[44]. V. Frinken and S. Uchida, "Deep BLSTM neural networks for unconstrained continuous handwritten text recognition," 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 911-915, doi: 10.1109/ICDAR.2015.7333894.

[45]. Shuohao Li, Anqi Han, Xu Chen, Xiaoqing Yin, and Jun Zhang "Review network for scene text recognition," Journal of Electronic Imaging 26(5), 053023 (17 October 2017). https://doi.org/10.1117/1.JEI.26.5.053023

[46]. 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)

[47]. Source OCR: Tesseract OCR [Computer software manual]. Retrieved from https://github.com/ tesseract-ocr/. Accessed 14 Aug 2018
[48]. Hellwig, O.: Indsenz ocr [Computer software manual]. Retrieved from http://www.indsenz.com/int/index.php. Accessed 14 Aug 2018
[49]. tdil: eaksharaya [Computer software manual] (2018). Retrieved from http://tdil-dc.in/eocr/index.html. Accessed 14 Aug 2018
[50]. Technology development for Indian languages [Computer software manual]. Retrieved from http://tdil.meity.gov.in/. Accessed 26 Jan 2019