# An efficient Multi Lingual Optical Character Recognition system for Indian languages through use of Bharati Script

Chandra Sekhar Vorugunti[1], Srinivasa Chakravarthy[2] , Viswanath Pulabaigari[1]

[1] Indian Institute of Information Technology-SriCity,Chittoor
[2] Indian Institute of Technology-Madras

[1]chandrasekhar.v@iiits.in
[1]viswanath.p@iiits.in
[2]schakra@ee.iitm.ac.in

**Abstract.** Optical character recognition performs a critical part in interpreting videos and documents. Document specific issues like low image quality, distortions, compo-site background, noise etc. and language specific issues like cursive connectivity among the characters etc. makes OCR challenging and erroneous for Indian lan-guages. The language specific challenges can be overcome by computing the script-based features and can achieve better accuracy. Computing the script based invariant features and patterns is computationally complex and error prone. In this background, we put forward Bharathi script* based OCR system in which the inherent drawbacks of Indian scripts i.e. Hindi, Tamil, Telugu etc. are eliminated. The proposed OCR model has been tested on a synthetic dataset of documents of Bharathi script (in which Hindi scripts are converted to Bharathi script). Thor-ough experimental analysis with varied levels of noise confirms the promising re-sults of character recognition accuracy of the proposed OCR model which out-performs the state-of-the-art OCR systems for Indian scripts. The proposed mod-el achieves 76.70% with test documents consists of 50% noise and 99.98% with test documents of 0% noise.

**Keywords:** Optical character recognition, Convolutional neural network, Deep learning, Indic script recognition.

## 1 Introduction

Optical character recognition (OCR) consists of identifying handwritten as well as the printed characters from a digital document, produced by scanning a hardcopy, converting the characters into suitable symbolic code thereby producing an editable document [1,2,10,16,22]. Variations in the physical characteristics of document im-ages, low-quality of the text, noise make the OCR a challenging task. Due to the efficiency in managing voluminous information, OCR has important applications in postal services, banking, office automatization initiatives [3,6,8,34].

*www.bharatiscript.com

Recent advancements in computing capacity and machine learning techniques results in increasing usage of OCR in a developing country like India. In Roman script which is used to express English and other West European languages, there are only 26 characters. Any word is a string of these isolated symbols.Unlike English, most Indic scripts are *abugida* i.e. writing systems where the vowels are inscribed as diacritics on the consonants and a vowel is not explicitly written when it present next to a consonant in a word. This sequence of diacritics with consonants is termed a composite character or samyuktakshar. A consonant can combine with both each of the vowels and with other consonants of the writing system to form ligatures. Therefore the glyphs representing vowels and consonants are amalgamated according to complex rules of orthography to form new characters. For this reason, a typical Indic script (with the exception of Tamil) has of the order of 10,000 characters. These features make Indic scripts complex, posing significant challenges to development of language related technologies like OCR [5,11,13,20,21].

**Fig. 1.** The three tier structure of a Bharati akshara, Dhe (दे)

चित्रकूट के घाट पर भइ संतन की भीर।
तुलसिदास चंदन घिसें तिलक देत रघुबीर॥
आनन्दकानने ह्यस्मिंजंगमस्तुलसीतरुः।
कवितामंजरी भाति रामभ्रमरभूषिता॥

ఉప్పుగప్పురంబు నొక్కపోలికనుండు
చూడచూడ రుచుల జాడవేరు
పురుషులందు పుణ్య పురుషులువేరయ
విశ్వదాభిరామ వినుర వేమ

சிற்குணத்தர் தெரிவு அரு நல்நிலை
எற்கு உணர்த்த அரிது; எண்ணிய
முற்ற குணத்தவரே முதலோர்; அவர்
நற்குணக் கடல் ஆடுதல் நன்றுஅரோ.

**Fig. 2.** The representation of Hindi, Telugu and Tamil text in Bharati script.

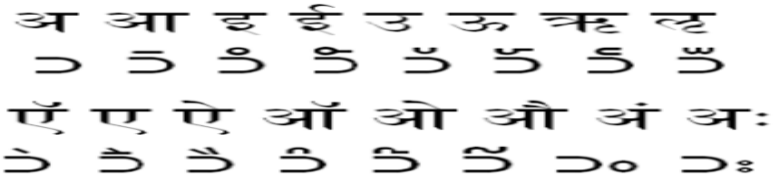


**Fig. 3.** Devanagari vowels (odd numbered row) and corresponding Bharati vowels (even numbered rows)

The recently proposed Bharati script [41], a novel unified Indic script that can be used to express most major Indian languages, offers some promise in terms of language technology development. The simplicity of its glyphs and lucid and logical compositionality makes it an ideal candidate for OCR development. Like most Indic characters, a typical Bharati character has a three- tier structure (upper, base and lower) (fig. 1, fig 3). The three tiers are disconnected and are clearly segmentable in OCR by connected component analysis. Most of the glyphs are simple and can be written, for example, in a single stroke without lifting the hand. The glyphs in the upper level always denote the vowel modifier; this level is empty if there is no vowel modifier and the implicit vowel is 'a.' The base (or middle) level always denotes the main consonant. The lower level has diacritics that modify the main consonant present in the base level. For example, of the base level has consonant 'ta', addition of a certain lower level glyph may convert the consonant 'ta' to 'da'. A single Bharati character can have only a single consonant and a vowel. Composite characters, of the type, say, CCV, are expressed as two Bharati characters: first character = C + <halant> ; second character = CV. By virtue of this simplifying feature the number of characters in Bharati script are much smaller than in current Indic script. In fact, using only about 40 glyphs, all the Bharati characters can be composed, which in turn can be used to express the tens of thousands of characters of various Indic scripts.

In this paper we present an OCR system for recognizing Bharati characters. As depicted Fig 2, the advantage of this system is that it can serve as a common OCR for most major Indian languages since Bharati can be used as a common script to express them. By designing Bharati characters as an additional font (the *NavBharati* fonts) for several of the major Indian languages, we can directly convert Indian language documents into Bharati script (fig. 2)

The proposed OCR system is tested on Hindi document images of expressed in Bharati script. Deep learning methods are used in this OCR system. Individual glyphs located in the three tiers of Bharati characters are recognized by three separate Convolutional Neural Networks (CNNs). Outputs of the three CNNs are combined using a set of rules thereby converting the original document image into Unicode. The proposed system yields close to 100% performance on noise-free documents.

## 2 Related Work

Earliest OCR systems in Hindi/Devnagari can be traced back to the '90s. The first of such models was proposed by Pal et al [14]. In their Devanagri OCR model, the structural, template features are retrieved from the documents, and a tree classifier is used to recognize the characters and achieved 95.19% recognition accuracy. Subsequently, other models have been proposed which are based on Center Distance Based features, Cut based features, Neighborhood counts based features [15-20]. Later, Ukil et al [21,22] have proposed an Indic OCR model for all the major languagges using CNN

based feature extraction. In which, conventional spatial domain representation, multi-level 2D discrete Haar wavelet transform are extracted for OCR and achieved 94.73% recognition accuracy. Chaudhuri et al [9] proposed the Hindi OCR models based on fuzzy multi-layer perceptron (FMLP), fuzzy Markov random fields (FMRF) and fuzzy support vector machines (FSVM) and achieved significant recognition accuracies 92%.

Models have been proposed based on Gaussian Mixture Models for OCR in Telugu[14], Malayalam, Gujarati and Hindi; these models surpass the traditional models based on SVM [5] on critical metrics like F-Score, recall, precision. Recently Parui et al. [4], Mahmoud et al. [6], Amin et al [23], Kundu et al. [24], have proposed Indic OCR models adopting Markov models of first and second orders, and Hidden Markov Models (HMM). These models used network parameters derived using statistical techniques as feature values and achieved respectable accuracy of 80.2%, 98.21%, 81% and 85% respectively in Hindi OCR. However, the OCR models based on HMM or the Markov models suffer from the inherent drawback of the requirement of a large number of training samples.

More recently, application of deep learning-based approaches to Indic OCR have pushed up recognition accuracies considerably. [1,3,13,15,18,19]. Rohit et al [15] proposed an LSTM with a delay, for mutual learning of language models and error patterns and shown that their proposed model is strong at detecting errors in Indic OCR with the recognition accuracies of 93.6%.

Recently notable work has been proposed for Indic OCR in the literature. Deep Neural Network (DNN) based models such as Denoising auto-encoders [29,30], LSTM based models [15,25,34], multi-column CNN based models [3,18,25,26, 28,41], BLSTM based memory networks [7, 26] have been proposed for the recognition of handwritten or printed characters and digits. Ray et al [26] proposed an OCR model for Oriya language using deep Bidirectional Long Short Term Memory (BLSTM) based Recurrent Neural Network and achieves 4.18% CER and 12.11% WER. In 2018, Wang et al [43] proposed an Indic OCR model for documents with low resolution. LSTM and RNN models are used to perform segmentation and retrieval of plurality of text lines from the document images. Pramanik et al [32] proposed a shape decomposition-based classification model for Bangla OCR, which achieves best recognition accuracy of 88.74% by reducing the number of classes to be recognized.

Recently, Ankan et al [42], proposed an innovative method which employed fusion techniques that comprises deriving of global and local features from image patches using CNN-LSTM framework and weighting them dynamically for character recognition and achieved 96% accuracy.

## 3 Proposed Work

As discussed above, the Bharathi characters have the intrinsic advantage of a clear separation among the upper, the base and the lower segments, a simplifying feature absent in other Indic scripts like for example Devanagari. In Devanagari, both vowel and consonant modifiers are connected to the main consonant following complex rules of ligature, a feature that poses significant challenges for OCR in Devanagari.

In this work, we propose an OCR system for Hindi documents expressed in Bharati script.

The remaining of the paper is structured as follows: Section 3.1 presents our CNN based model architecture. Thorough experimental analysis of our proposed model and comparison with recently proposed models are presented in section 4. Finally, in section 5, we conclude the paper with a defined future direction.

### 3.1 Proposed CNN architecture for OCR

In this segment, we do a thorough explanation of the proposed OCR-CNN architecture, preprocessing and data augmentation techniques used for Bharathi OCR.

**CNN Architecture.**
Deep Convolutional Neural Networks (CNN) are multilayer neural networks consists of several convolutional layers interleaved by pooling layers, which down sample the images before feeding to subsequent layers. We used Rectified Linear Units (ReLU) as the activation function in all the convolutional and fully connected layers. To adjust and optimize the weights of the model during backpropagation, the Gradient descent is used and a differentiable loss function is chosen. We have chosen 'adam' as an optimizer. Apart from the convolution layers, dropouts and pooling layers are added to augment the performance of the model. We used batch optimization in training. The comprehensive list of CNN parameters is shown in Table 1.

The figure 4, depicts our model for training 'Base' segments of Bhrathi character. The first convolutional layer filter the 28 × 28 input image with 128 kernels of size 3 × 3 with a padding of 1 pixels. Padding is essential in order to convolve the filter from the very first pixel of the input image. The second convolutional layer takes as input the (zero center normalization and pooled) output of the first convolutional layer and filters it with 128 kernels of size 3 × 3 with a padding of 1 pixels. The third layer is a max pooling layer has pool size 2 × 2 and stride of two. Stride indicates the number of steps to be skipped for the next convolution and pooling operations. A dropout of 25% is applied in the max-pooling layer. The first fully connected layer has 128 neurons, whereas the second fully connected layer has 11/17/8 neurons depends the type of the character trained i.e. upper, base or lower. A drop out of 50% is applied to the second

fully connected layer. Figure 4 demonstrates the flow diagram of the proposed approach.

The weights are randomly initialized and the biases equal to zero. We trained the model using Adam for 20 epochs. Our entire framework is implemented using MATLAB deep learning libraries. The training was done using a GeForce GTX 1070 and a TITAN X Pascal GPU, and it took approximately 3 hours to run each CNN.

**Table 1.** Our proposed model architecture details.

| Layer | Size | Parameters |
|---|---|---|
| Image Input | 28 × 28×1 | images with 'zero center' normalization |
| Convolutiond Layer (3,128,'Padding',1) | 128× 3 × 3 | stride [0 0] Padding [1 1 1 1 ] |
| ReLU | | ReLU |
| Convolutiond Layer (3,128,'Padding',1) | 128× 3 × 3 | stride [0 0] Padding [1 1 1 1 ] |
| ReLU | | ReLU |
| Max Pooling | | 2x2 max pooling with stride [2 2] and padding [0 0 0 0] |
| Dropout | 25% dropout | |
| Fully Connected | 128 fully connected layer | |
| ReLU | | ReLU |
| Dropout | 50% dropout | |
| Fully Connected | 16/20/8 fully connected layer | |
| Softmax | | softmax |
| Classification Output | | crossentropyex |

## Preprocessing.

Since Bharathi script is a novel script, no commercial datasets for Bharathi characters are readily available. Therefore, we developed our own synthetic Bharathi character dataset. As discussed above, Bharathi character is divided into three independent segments ‘upper’, ‘base’, ‘lower’. There are totally 11 upper, 17 base and 8 lower glyphs in Bharati system.

In line with the restriction on the size of the input images for batch training a convolutional neural network, we resize all the images (upper,base,lower) to a fixed size of 28 × 28. Afterwards, we convert the training images to gray scale images so that the background pixels have 0 values

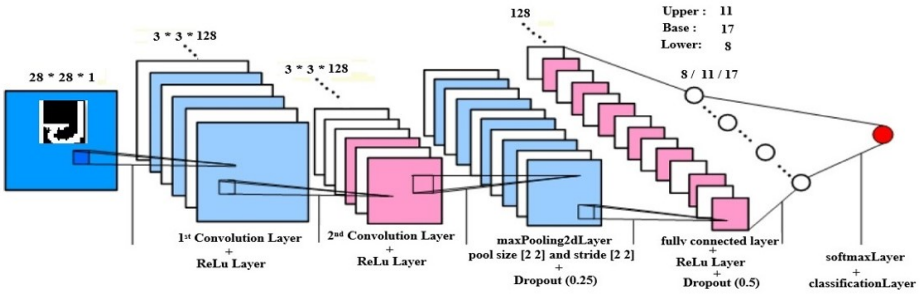


**Fig. 4.** Illustration of the proposed CNN Model architecture with all significant components highlighted.

**Data Augmentation.**
To deal with the deficiency of sufficient training data for Bharathi script, we augmented each train image by translation through all the eight directions like horizontal, vertical, diagonal, etc. Also, to study the impact of noise on character recognition, we expanded the data set by adding gaussian and the salt and pepper noise.

## 4 Experiments

Our model consists of three CNNs corresponding to upper, base and lower segments of Bharathi character (fig 1). During training phase of our model, we train each of the three CNN using our synthetic dataset which consists of 6400 images, each of size 28 × 28 for each Bharathi component (pertaining to upper, base and lower segments). Bharathi character set consists of 11 upper, 17 base and 8 lower characters. Therefore a total 281600 = 6400 * 44 images are available in the dataset, out of which 4000 were used for training and 2400 were used for validation of each of the upper, base and lower segments. To test the proposed CNN based OCR model, we used test image documents consists of Bharathi characters as shown in appendix. During testing, each Bharathi character is extracted from the input test document, by segmenting the document image first into individual lines, and the lines into individual words and characters respectively. Further, the extracted Bharathi character is split into upper, base and lower segments. In Bharathi, the base segment is mandatory, while the upper and lower segments are optional. Each extracted segment is given as input to the corresponding CNN and the classification result is noted.

### 4.1 Experimental Protocol

We tested the model performance in OCR with increasing levels of noise ranging from 0% (no noise) , 5%, 10%, 20%, 30%, 40%, 50% noise levels added to the character images. We trained the model with images consists of 0% and 2% noise. As depicted in table 2, the average optical character recognition accuracy is decreasing with the increasing values of noise. The table illustrates that even though the classification accuracy is not affected with 0-20% of noise, from 30-50% of noise, there is steep fall in the classification accuracy. The ‘Base’ segments have shown a steep drop in accuracy with the increased noise, varying between 40-50%. Even in classifying the images with 50% of noise, our proposed model is outperforming all the recent proposed models in the literature with 76.70% (avg of 84.7, 48.22, 97.19 i.e. classification accuracy of lower, base and upper cnn with a test document of 50% noise) of classification accuracy.

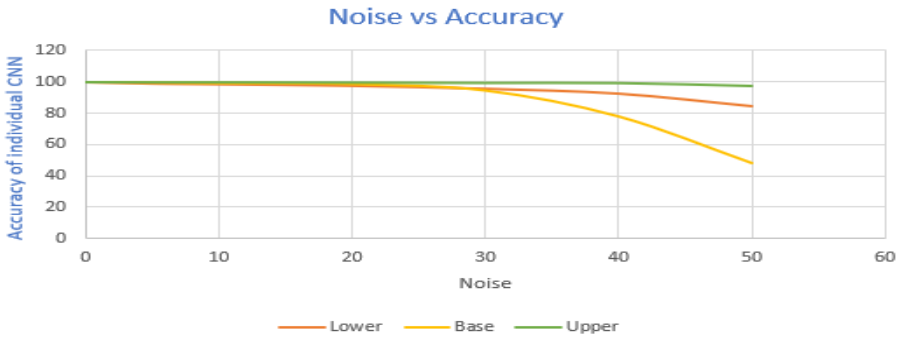
**Table 2.** The classification accuracies of Individual CNN's (Lower/Base/Upper) in recognizing the lower, base and upper characters with increasing noise percentage in the test documents images (File 1 to File 14).

| | | Lower | | | | | | | | Base | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Noise % | 0 | 5 | 10 | 20 | 30 | 40 | 50 | | Noise | 0 | 5 | 10 | 20 | 30 | 40 | 50 |
| File 1 | 99.98 | 100 | 100 | 98.17 | 94.5 | 86.24 | 80.73 | | File 1 | 99.99 | 99.31 | 99.31 | 98.79 | 94.98 | 77.51 | 44.29 |
| File 2 | 100 | 92.86 | 92.86 | 92.86 | 85.71 | 100 | 92.86 | | File 2 | 100 | 99.11 | 99.11 | 99.11 | 93.81 | 75.22 | 53.98 |
| File 3 | 99.45 | 96.4 | 95.5 | 96.4 | 94.6 | 90.99 | 79.3 | | File 3 | 99.99 | 99.83 | 99.18 | 99.34 | 94.73 | 79.07 | 50.25 |
| File 4 | 99.94 | 99.23 | 96.15 | 98.46 | 92.31 | 90.77 | 83.85 | | File 4 | 100 | 99.81 | 99.63 | 98.51 | 94.41 | 74.3 | 48.6 |
| File 5 | 100 | 100 | 100 | 90 | 100 | 90 | 90 | | File 5 | 100 | 100 | 100 | 100 | 100 | 81.94 | 52.78 |
| File 6 | 99.98 | 99.34 | 98.01 | 96.69 | 95.36 | 93.38 | 87.42 | | File 6 | 99.99 | 99.55 | 99.1 | 98.64 | 94.43 | 74.85 | 46.08 |
| File 7 | 99.99 | 99.44 | 100 | 98.88 | 98.31 | 95.51 | 89.33 | | File 7 | 100 | 99.62 | 99.75 | 98.5 | 93.61 | 73.81 | 42.73 |
| File 8 | 100 | 100 | 100 | 100 | 90.91 | 90.91 | 77.27 | | File 8 | 100 | 100 | 97.44 | 96.15 | 86.54 | 66.03 | 41.03 |
| File 9 | 99.98 | 100 | 100 | 100 | 99.26 | 94.81 | 80 | | File 9 | 100 | 99.83 | 99.83 | 98.67 | 94.35 | 93.69 | 41.69 |
| File 10 | 99.98 | 100 | 100 | 98.67 | 97.33 | 92.67 | 83.33 | | File 10 | 100 | 99.84 | 99.84 | 99.53 | 95 | 90.78 | 45 |
| File 11 | 100 | 100 | 100 | 99.98 | 100 | 92.68 | 92.68 | | File 11 | 100 | 100 | 100 | 98.77 | 97.53 | 77.78 | 52.47 |
| File 12 | 99.95 | 100 | 100 | 98.95 | 97.89 | 94.74 | 82.11 | | File 12 | 100 | 99.82 | 99.45 | 99.08 | 96.7 | 78.88 | 64.59 |
| File 13 | 100 | 98.9 | 98.9 | 98.9 | 97.89 | 95.6 | 87.91 | | File 13 | 100 | 99.82 | 99.1 | 98.38 | 94.79 | 77.38 | 47.76 |
| File 14 | 99.97 | 100 | 98.77 | 98.77 | 96.3 | 88.89 | 79.01 | | File 14 | 100 | 100 | 99.75 | 99 | 94.74 | 71.93 | 43.86 |
| Avg Acc | 99.944 | 99.012 | 98.585 | 97.623 | 95.740 | 92.656 | 84.7 | | Avg Acc | 99.99 | 99.75 | 99.392 | 98.747 | 94.687 | 78.083 | 48.2221 |

| | | Upper | | | | | |
|---|---|---|---|---|---|---|---|
| Noise | 0 | 5 | 10 | 20 | 30 | 40 | 50 |
| File 1 | 100 | 99.7 | 99.7 | 99.4 | 98.8 | 99.1 | 98.2 |
| File 2 | 100 | 100 | 100 | 100 | 100 | 100 | 98.39 |
| File 3 | 100 | 99.68 | 99.68 | 99.05 | 99.68 | 98.42 | 96.2 |
| File 4 | 100 | 100 | 100 | 100 | 100 | 100 | 99.35 |
| File 5 | 100 | 100 | 100 | 100 | 100 | 100 | 97.5 |
| File 6 | 100 | 99.25 | 98.75 | 99.25 | 99 | 98.75 | 96.75 |
| File 7 | 100 | 100 | 99.79 | 100 | 99.36 | 99.57 | 98.5 |
| File 8 | 100 | 100 | 100 | 100 | 100 | 100 | 92.93 |
| File 9 | 100 | 99.72 | 100 | 100 | 99.72 | 100 | 97.18 |
| File 10 | 100 | 100 | 100 | 99.73 | 99.46 | 99.19 | 95.12 |
| File 11 | 100 | 100 | 100 | 98.98 | 97.96 | 97.96 | 95.92 |
| File 12 | 100 | 100 | 100 | 100 | 100 | 99.36 | 98.72 |
| File 13 | 100 | 99.69 | 100 | 99.69 | 99.69 | 99.69 | 98.14 |
| File 14 | 100 | 100 | 100 | 99.56 | 98.67 | 98.67 | 97.79 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Avg Accuracy | 100.00 | 99.86 | 99.8514 | 99.69 | 99.4528 | 99.33 | 97.19 |

**Fig. 5.** Plot of classification accuracies of individual CNN (Upper/Base/Lower) with increased levels of noise from 0% to 50% in the test document images.




In line with the literature, to evaluate our proposed OCR model and compare it with the recent proposed models, we use the standard evaluation metrics i.e Character Error Rate (CER) and Word Error Rate (WER).

CER and WER are described as (CT: classified text and GT: ground truth text in the below equation)-

$$\text{CER} = \frac{\sum EditDistance(CT,GT)}{\#of\ unicodes\ in\ the\ document} \quad (1)$$

$$\text{WER} = \frac{\sum EditDistance(CT,GT)}{\#of\ words\ in\ the\ document} \quad (2)$$

i.e. the sum of substitutions, deletions and insertions in terms of unicodes essential to convert CT to GT, divided by the amount of unicodes in the ground truth (input test file). WER is defined as the average count of words incorrectly classified.

Note that although Bharati script does not have a separate Unicode, since Bharati characters are an alternative system of expression for Indic languages, the output of the Bharati document in the current case can be expressed in Unicode. Table 3 shows how the CER increases with the increased levels of noise in the test document images. CER at 50% noise is 0.326. Similarly, Table 4 summarizes how the WER increases with the increased levels of noise in the test documents. WER at 50% of noise is 1.599. Figure 6, illustrates that both the errors CER and WER are slowly increasing till 20% of the noise. After that there is steep rise in the error curves.

Table 5 confirms that the proposed model outperforms the other recent proposals in terms of word level and character level classification error. Our proposed model classification error with 50% of noise in the test document is much less compared to classification error of other models with 0% of noise in the test document. The model proposed by Chaudhuri et al. [9] is recorded less CER compared to our model. The CER value of the model proposed by Chaudhuri et al. [9] is 0.2 at 0% of noise, whereas CER

of our proposed model is 0 and 0.33 at 0% and 50% of noise in the test document respectively.

Based on these experimental analyses we can confirm that the proposed method of converting the Devanagari/Hindi script to Bhrathi script and performing the OCR using CNN model trained on Bharathi script is yielding results that are far superior to those reported in the literature.

**Table 3.** The Character Error Rate of model with increased levels of noise in the test documents

| File ID | # of Unicodes | Noise 5% | CER | Noise 10% | CER | Noise 20% | CER | Noise 30% | CER | Noise 40% | CER | Noise 50% | CER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1020 | 5 | 0.005 | 5 | 0.005 | 11 | 0.011 | 39 | 0.038 | 148 | 0.145 | 349 | 0.342 |
| 2 | 251 | 2 | 0.008 | 2 | 0.008 | 2 | 0.008 | 9 | 0.036 | 28 | 0.112 | 55 | 0.219 |
| 3 | 1034 | 6 | 0.006 | 11 | 0.011 | 11 | 0.011 | 39 | 0.038 | 142 | 0.137 | 337 | 0.326 |
| 4 | 975 | 2 | 0.002 | 7 | 0.007 | 10 | 0.01 | 40 | 0.041 | 150 | 0.154 | 299 | 0.307 |
| 5 | 121 | 0 | 0 | 0 | 0 | 1 | 0.008 | 0 | 0 | 14 | 0.116 | 37 | 0.306 |
| 6 | 1215 | 7 | 0.006 | 14 | 0.012 | 17 | 0.014 | 48 | 0.04 | 182 | 0.15 | 390 | 0.321 |
| 7 | 976 | 4 | 0.004 | 3 | 0.003 | 14 | 0.014 | 56 | 0.057 | 217 | 0.222 | 483 | 0.495 |
| 8 | 271 | 0 | 0 | 4 | 0.015 | 6 | 0.022 | 23 | 0.085 | 55 | 0.203 | 104 | 0.384 |
| 9 | 1091 | 2 | 0.002 | 1 | 0.001 | 8 | 0.007 | 36 | 0.033 | 45 | 0.041 | 388 | 0.356 |
| 10 | 1159 | 1 | 0.001 | 1 | 0.001 | 6 | 0.005 | 38 | 0.033 | 73 | 0.063 | 395 | 0.341 |
| 11 | 301 | 0 | 0 | 0 | 0 | 4 | 0.013 | 6 | 0.02 | 41 | 0.136 | 84 | 0.279 |
| 12 | 953 | 1 | 0.001 | 3 | 0.003 | 6 | 0.006 | 20 | 0.021 | 121 | 0.127 | 214 | 0.225 |
| 13 | 971 | 3 | 0.003 | 6 | 0.006 | 11 | 0.011 | 32 | 0.033 | 131 | 0.135 | 308 | 0.317 |
| 14 | 707 | 0 | 0 | 2 | 0.003 | 6 | 0.008 | 27 | 0.038 | 124 | 0.175 | 246 | 0.348 |
| avg | | | 0.002673 | | 0.005284 | | 0.010736 | | 0.036603 | | 0.13688 | | 0.326036 |

**Table 4.** The Word Error Rate of model with increased levels of noise in the test documents.

| File ID | # of Unicodes | Noise 5% | WER | Noise 10% | WER | Noise 20% | WER | Noise 30% | WER | Noise 40% | WER | Noise 50% | WER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 246 | 5 | 0.02 | 5 | 0.02 | 11 | 0.045 | 39 | 0.159 | 148 | 0.602 | 349 | 1.419 |
| 2 | 44 | 2 | 0.045 | 2 | 0.045 | 2 | 0.045 | 9 | 0.205 | 28 | 0.636 | 55 | 1.25 |
| 3 | 259 | 6 | 0.023 | 11 | 0.042 | 11 | 0.042 | 39 | 0.151 | 142 | 0.548 | 337 | 1.301 |
| 4 | 240 | 2 | 0.008 | 7 | 0.029 | 10 | 0.042 | 40 | 0.167 | 150 | 0.625 | 299 | 1.246 |
| 5 | 33 | 0 | 0 | 0 | 0 | 1 | 0.03 | 0 | 0 | 14 | 0.424 | 37 | 1.121 |
| 6 | 271 | 7 | 0.026 | 14 | 0.052 | 17 | 0.063 | 48 | 0.177 | 182 | 0.672 | 390 | 1.439 |
| 7 | 312 | 4 | 0.013 | 3 | 0.01 | 14 | 0.045 | 56 | 0.179 | 217 | 0.696 | 483 | 1.548 |
| 8 | 320 | 0 | 0 | 4 | 0.013 | 6 | 0.019 | 23 | 0.072 | 55 | 0.172 | 104 | 0.325 |
| 9 | 59 | 2 | 0.034 | 1 | 0.017 | 8 | 0.136 | 36 | 0.61 | 45 | 0.763 | 388 | 6.576 |
| 10 | 245 | 1 | 0.004 | 1 | 0.004 | 6 | 0.024 | 38 | 0.155 | 73 | 0.298 | 395 | 1.612 |
| 11 | 60 | 0 | 0 | 0 | 0 | 4 | 0.067 | 6 | 0.1 | 41 | 0.683 | 84 | 1.4 |
| 12 | 232 | 1 | 0.004 | 3 | 0.013 | 6 | 0.026 | 20 | 0.086 | 121 | 0.522 | 214 | 0.922 |
| 13 | 227 | 3 | 0.013 | 6 | 0.026 | 11 | 0.048 | 32 | 0.141 | 131 | 0.577 | 308 | 1.357 |
| 14 | 282 | 0 | 0 | 2 | 0.007 | 6 | 0.021 | 27 | 0.096 | 124 | 0.44 | 246 | 0.872 |
| Average | | | 0.013674001 | | 0.01990565 | | 0.046665 | | 0.164072 | | 0.546917 | | 1.599228 |

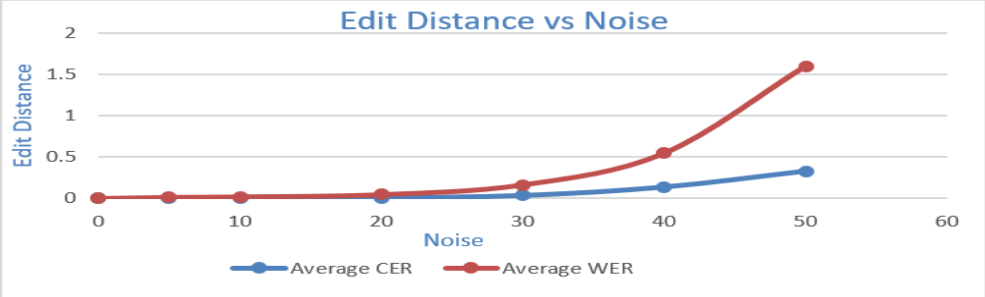



**Fig. 6.** Plot of edit distance between OCR word and corresponding ground truth word pairs.

**Table 5.** Performance evaluation of the proposed model with the recently proposed models in the literature.

| | WER (%) (Noise – 0%) | CER (%) (Noise-0%) |
|---|---|---|
| Verma et al [1] | - | 15 |
| Hanmandlu et al. [2] | - | 9.35 |
| Rojatkar et al. [3] | - | 2.38 |
| Parui et al [4] | - | 17.11 |
| Gyanendra et al. [5] | - | 10 |
| Arora et al. [8] | - | 7.2 |
| Chaudhuri et al. [9] | - | 0.2 |
| Deshpande et al. [10] | - | 18 |
| Sharma et al [11] | - | 19.64 |
| Deepti et al [12] | - | 8.6 |
| Sarkhel et al [13] | - | 4.82 |
| Pal et al [14] | - | 4.81 |
| Rohit[15] | - | 7.09 |
| Kartik et al [18] | 4.62 | 2.67 |
| Kartik et al [18] | 11.89 | 4.9 |
| Kartik et al [18] | 14.09 | 5.53 |
| Bappaditya et al [19] | - | 3.91 |
| Ritesh et al [20] | - | 4.58 |
| Shrawan et al [33] | - | 3.10 |
| Roy et al. [36] | 15.76 | - |
| Agnihotri et al. [37] | - | 14.22 |
| Shelke et al. [38] | - | 5.34 |
| Joshi et al. [39] | - | 12.59 |
| LeCun et al. [40] | - | 22.70 |
| Ray et al [26] | 12.11 | 4.18 |
| Proposed Model at Noise 50% | 1.6 | 0.33 |
| Proposed Model at Noise 0% | 0.002673 | 0.013674001 |

## 5 Conclusion

In this paper, we have proposed an efficient OCR model based on Convolutional Neural Networks (CNN) for Hindi documents expressed in Bharathi script. That is, instead of the native Devanagari script, and Hindi documents are expressed in Bharati script. The

OCR system applied to the Bharati-Hindi document images yielded significantly better performance on the original Devanagari documents. Due to underlying representational power of Bharathi character, unlike its predecessors for OCR, our model achieves excellent classification results. To demonstrate the advantage, we have conducted thorough experiments on the synthetic dataset. The experiments demonstrate a high level of accuracy in character recognition. Furthermore, the proposed model is tested against the test document images upto 50% noise and the recognition results of our model surpassed the state-of-the-art results. Our future work in this direction is to emphasis on the development of more enriched OCR models and large datasets for all the Indian languages.

## References


1. Verma, B.K.: Handwritten Hindi character recognition using multilayerperceptionand radial basis function neural networks. IEEE Int. Conf. Neural Networks, Perth, Australia, pp. 2111–2115. (1995).
2. Hanmandlu, M., Murthy, O. V. R., Madasu, V. K.: Fuzzy modelbasedrecognition of handwritten Hindi characters. In: Biennial Conf. Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, Glenelg, Australia, pp. 454–461.
3. Rojatkar, D. V., Chinchkhede, K.D., Sarate, G. G.: Handwritten devnagariconsonants recognition using MLPNN with fivefold cross validation. Int.Conf. Circuits, Power and Computing Technologies (ICCPCT), Nagercoil,India, pp. 1222–1226.
4. Parui, S.K., Shaw, B.: Offline handwritten Devanagari word recognition: an HMM based approach. In: Ghosh, A., De, R.K., Pal, S.K. (Eds.). Pattern recognition and machine intelligence (Springer PReMI, Berlin-Heidelberg,2007), Lecture Notes in Computer Science, pp. 528–535.( 2007).
5. Verma, G. K., Prasad, S., Kumar, P.: Handwritten Hindi character recognitionusing curvelet transform. In: Int. Conf. Information Systems for IndianLanguage, Patiala, India, 2011, pp. 224–227. (2011).
6. Mahmoud, S.: Recognition of writer-independent off-line handwritten Arabic (Indian) numerals using hidden Markov models.Sign. Process. 88 (4) 844–857.(2008).
7. Raman, J., Volkmar, F., Jawahar, C.V., Manmatha, R.: BLSTM Neural Network Based Word Retrieval for Hindi Documents. In: International Conference on Document Analysis and Recognition,Beijing, China.(2011).
8. Arora, S., Bhattacharjee, D., Nasipuri, M.:Combining multiple featureextraction techniques for handwritten devnagari character recognition. In: Thirdint.Conf. Industrial and Information Systems, Kharagpur, India, pp. 1–6. (2011).
9. Chaudhuri, A., Mandaviya, K., Badelia, P.: Optical characterrecognition systems for Hindi language. In: Studies in fuzziness and softcomputing (Springer, Cham,1st edn.), pp. 193–216. (2017).
10. Deshpande, P. S., Malik, L., Arora, S.: Fine classification and recognition of hand written devnagari characters with regular expressions and minimum editdistance method. J. Comput. 3, (5), pp. 11–17. (2008).
11. Sharma, N., Pal U., Kimura F.: Recognition of off-line handwrittendevnagari characters using quadratic classifier. In: Proc. Indian Conf. ComputerVision, Graphics and Image Processing, Madurai, India, pp. 805–816.

12. Khandja, D., Nain, N., Panwara, S.: Hybrid feature extraction algorithm forDevanagari script. In: ACM Trans. Asian Low Resour. Lang. Inf. Process., 15, (1), p. 2:1–2:10. (2015).
13. Bing Su, Xiaoqing Ding, Hao Wang, Ying Wu.: Discriminative Dimensionality Reduction for Multi-Dimensional Sequences. In: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol 40. (2018).
14. Pal, U., Wakabayashi, T., Kimura, F.: Comparative study of devnagarihandwritten character recognition using different feature and classifiers. In: Int.Conf. Document Analysis and Recognition, Barcelona, Spain, July 2009, pp.1111–1115. (2009).
15. Rohit, S., Devaraj, A., Parag, C., Ganesh, R., Mark, C., : Error Detection and Corrections in Indic OCR using LSTMs. In: 14th IAPR International Conference on Document Analysis and Recognition. (2017).
16. Parul, S., Sanjay, B. D. : Multilingual Character Segmentation and Recognition Schemes for Indian Document Images. In: IEEE Access,VOLUME XX. (2017).
17. Jayadevan, R., Satish, R. Kolhe, Pradeep M. Patil, Umapada Pal.: Offline Recognition of Devanagari Script: A Survey. In: IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 41, NO. 6. (2011).
18. Kartik, D., Praveen, K., Minesh, M., Jawahar, C.V.: Towards Accurate Handwritten Word Recognition for Hindi and Bangla. In: National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics.
19. Bappaditya, C., Bikash, S., Jayanta, A., Ujjwal, B., Swapan, K,P., Does Deeper Network Lead to Better Accuracy: A Case Study on Handwritten Devanagari Characters. In: 13th IAPR International Workshop on Document Analysis Systems (DAS). (2018).
20. Ritesh, S., Nibaran, D., Aritra, D., Mahantapas, K., Mita, N..: A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts.Pattern Recognition 71, 78–93.
21. Soumya, U., Swarnendu, G., Sk Md Obaidullah, Santosh, K.C., Kaushik, R., Nibaran Das.: Deep learning for word-level handwritten Indic script identification, arxiv 2018. (2018).
22. Sankaran, N., Jawahar, C.: Error Detection in Highly InflectionalLanguages in Document Analysis and Recognition. (ICDAR), 12thInternational Conference on, pp. 1135–1139. (2013).
23. Amin, A.: Off-line Arabic character recognition.Pattern Recognit. 31 (5) 517–530. (1998).
24. Kundu, A., He, Y., Bahl, P.: Recognition of handwritten word: first and second order hidden Markov model based approach.In: Computer Vision and Pat- tern Recognition, Proceedings CVPR'88., Computer Society Conference on, pp. 457–462. (1988).
25. Ul-Hasan, A., Bin Ahmed, S., Rashid, F., Shafait, F., Breuel, T.M.: Online printed urdu nastaleeq script recognition with bidirectional LSTM networks. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1061–1065. (2013).
26. Ray, A., Rajeswar, S., Chaudhury, S.: Text recognition using deep BLSTM networks.In: 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), pp. 1–6. (2015).
27. Roy, S., Das, N., Kundu, M., Nasipuri, M.: Handwritten isolated bangla compound character recognition: a new benchmark using a novel deep learning approach.Pattern Recognit. Lett. 90 . 15–21. (2017).
28. Cire D¸san, Meier U.: Multi-column deep neural networks for online handwrit- ten Chinese character classification.In: International Joint Conference on Neural Networks (IJCNN), pp. 1–6. (2015).
29. Pal, A., Pawar, J.D.: Recognition of online handwritten Bangla characters using hierarchical system with denoising autoencoders.In: Computation of Power, Energy Information and Commuincation (ICCPEIC), 2015 International Conference on, pp. 47–51. (2015).

30. Pal, A.: Bengali handwritten numeric character recognition using denoising autoencoders.In: Engineering and Technology (ICETECH), 2015 IEEE International Conference on, pp. 1–6. (2015).
31. Saikat, R., Nibaran, D., Mahantapas, K., Mita, N., : Handwritten isolated Banga compound character recognition: A new benchmark using a novel deep learning approach .Pattern Recognition Letters 90 . 15–21. (2017).
32. Rahul, P., Soumen, B., : Shape decomposition-based handwritten compound character recognition for Bangla OCR. Journal of Visual Communication and Image Representation 50 123–134. (2018).
33. Shrawan, R., Shloak, G., Basant, A., : Devanagri character recognition model using deep convolution neural network. In: Journal of Statistics and Management Systems Volume 21, Issue 4. (2018).
34. Ankan, K.B., Aishik, K., Ayan, K.B., Abir, B., Partha, P. R., Umapada, P., : Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. Pattern Recognition 85.172–184. (2019).
35. Meduri, A.,Navneet, G.,: Optical Character Recognition for Sanskrit using Convolution Neural Networks. In: 13th IAPR International Workshop on Document Analysis Systems. (2018).
36. Roy, P.P., Bhunia, A.K., Das, A., Dey, P., Pal, U.: Hmm-based indic handwritten word recognition using zone segmentation. PR. (2016).
37. Agnihotri, V.P.: Offline handwritten devanagari script recognition. In: Int. J. Inf. Technol. Comput. Sci. 4 (8) (2012) 37. (2012).
38. Shelke, S., Apte, S.: A novel multi-feature multi-classifier scheme for unconstrained handwritten devanagari character recognition.In: 2010 12th International Conference on Frontiers in Handwriting Recognition, pp. 215–219. (2010).
39. Joshi, N., Sita, G., Ramakrishnan, A.G, Deepu, V., Madhvanath, S.: Machine recognition of online handwritten Devanagari characters.In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05), 2, pp. 1156–1160 . (2005).
40. LeCun Y, Bengio Y. Word-level training of a handwritten word recognizer based on convolutional neural networks.In: International Conference on Pattern Recognition, pp. 88–92.
41. Manali, N., Srinivasa, C.V., .: A comparative study of complexity of handwritten Bharati characters with that of major Indian scripts.In: International Joint Conference on Neural Networks (IJCNN), USA. (2017).
42. Ankan, K.B., Aishik, K.,Abir, B.,Partha P. R., Umapada, P., : Script identification in natural scene image and video frames using an attention based Convolutional-LSTM network. In: Elsevier journal of Pattern Recognition, vol 85, pp: 172–184. (2019).
43. Shuai, W., Maneesh, K.S., : Systems and Methods for Optical Character Recognition for Low-Resolution Documents. Patent: US20180101726A1.