RESEARCHERID
THOMSON REUTERS

[Banger * *et al.*, 7(5): May, 2018]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HINDI LANGUAGE RECOGNITION SYSTEM USING NEURAL NETWORKS

R.Ankush Banger[1], Mansi Singh[1], Kirnesh Sharma[1], Satvik Singla[1],Mrs.Shikha Rastogi[2]
1(Student, IT Department, Bharati Vidyapeeth's College Of Engineering, New Delhi)
2(Assistant Professor, IT Department, Bharati Vidyapeeth's College Of Engineering, New Delhi)

DOI: 10.5281/zenodo.1241407

ABSTRACT

In this paper, we propose a recognition scheme for the Indian script of Hindi. Recognition accuracy of Hindi script is not yet comparable to its Roman counterparts. This is mainly due to the complexity of the script, writing style etc. Our solution uses a Recurrent Neural Network known as Bidirectional Long Short Term Memory (BLSTM). Our approach does not require word to character segmentation, which is one of the most common reason for high word error rate. We report a reduction of more than 20% in word error rate and over 9% reduction in character error rate while comparing with the best available OCR system.

Keywords: BLSTM, Word recognition, Hindi, OCR.

I. INTRODUCTION

Most of the present day Optical Character Recognizers (OCR) show impressive results for a wide range of documents in Roman scripts. Past few years have seen considerable interest in developing similar OCR systems for Indic scripts [6]. Several improvements have been made in this frontier which resulted in significant advancement in techniques and improvements in results [1, 6]. Still the accuracies drop heavily with degradations. Most of the traditional OCRs use character segmentation to extract symbols and recognize them using a classifier. However, such methods fail in poor quality documents due to the presence of cuts and merges. Degradation scan arise due to multiple sources like poor quality of ink, low spacing between characters, document age etc. Considerable effort was made to find a solution to this problem at character level [10]. However, trying to find a solution at character level may not be the ideal approach,as such methods may fail when it comes to large diverse documents containing variation in font, degradation etc.

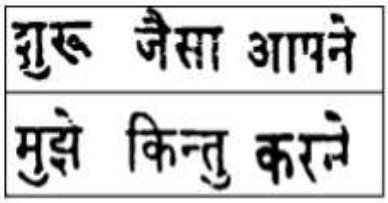


Figure 1. Some word examples which are being correctly identified by our method. Presence of degradation prevent traditional OCR to recognize them correctly.

We believe that recognition of degraded characters requires more than just the knowledge of which connected components can be joined/split to form a valid component. Using contextual information can be a good way to solve such problems. A word,when looked as a single component, rather than a collection of characters, provide more information than what an isolated character can provide. Recognition at word level is not new. There have been several other word recognition systems in the past which tried to address this issue [11, 3, 9]. We propose to recognize Hindi words by using a neural network classifier known as Bidirectional Long Short-Term Memory (BLSTM) Neural network [7]. BLSTM based word recognition have been proven quite successful in recognizing a variety of documents including handwritten documents [4]. We have used the same neural network for word retrieval for Hindi documents [8]. In this work, we extend this towards recognition. We also used a script-dependent segmentation module to obtain the word images from pages. This was needed largely

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

due to the presence of complex, multicolumn pages. Our Hindi layout module took page images as input and generated a set of word images.

II. WORD RECOGNITION USING BLSTM

We recognize Hindi text based on a variant of Recurrent Neural Networks, known as Bidirectional Long Short-Term Memory (BLSTM) neural network [7]. The BLSTM consists of hidden layers which are made up of the so-called long short-term memory (LSTM).Such a solution enables the network to predict the label sequence based on both the past and future context of the element. This is done by making use of two hidden layers. One to process the input sequence forward while other processes it backwards. Similar system has been used in word recognition and retrieval in the past [8]. Some of the advantages of the system are mentioned in [5]. The data is recognized at word level instead of character level so as to overcome the issues associated with degraded data while taking into account the context associated within a word. The problem of vanishing gradient, i.e. error gradients vanish exponentially with the size of the time lag between important events, has been addressed in the BLSTM network by using the LSTM hidden layers. The LSTM hidden layers consist of recurrently connected subnets,called memory blocks. Each block consists of a set of internal units whose activation is controlled by three multiplicative gates: the input gate, forget gate and output gate. Information can be stored and accessed over a long period of time using these gates. More details are present in [7]. The words written in Hindi script are connected by a head line, known as shirorekha. Vowel modifiers or matras can be made by combining a vowel with a consonant. The script also allows joining of multiple vowels and consonants to form a compound character. This results in having the number of unique characters anywhere between 300-800. This number depends on the font and the rendering scheme used. To reduce the number of unique classes, a commonly followed approach is to remove the shirorekha and recognize the characters and then recombine those at the end. This is referred to as zoning [2].

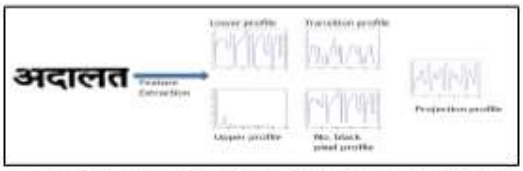


Figure 2. A sample word image and the five features extracted from it.

Our system is robust enough to handle the large number of classes and hence, we do not perform zoning of words. In this regard, our present work is considerably different from [8]. The number of unique class labels stand at 685 in our case. For every word, we extract 5 different features from a vertical strip of uniform width, using a sliding window. The features extracted are (a) the lower profile, (b)the upper profile,(c)the ink-background transitions, (d) the number of black pixels, and (e) the span of the foreground pixels. The upper and lower profiles measure the distance of the top and bottom foreground pixel from the respective baselines. Ink-background transitions measures the number of transitions from Ink to background and reverse. The number of black pixels provides the information about the density of ink in the vertical stripe. A sample word and its corresponding feature sequences are shown in Figure 2. We represent the word image as a sequence of feature vectors of dimension d. BLSTM neural networks contain one node for each of the features in the input layer. Thus the network has d input layer nodes. Each node in the input layer is connected with two separate hidden layers, one of which processes the input sequence of features forward, while the other processes it backward. Both hidden layers are connected to the same output layer. The output layer sums up the values which comes from both forward and backward hidden layers. Most of the Recurrent Neural Networks(RNN) require pre-segmented input label with a separate target for every label. However, such a mechanism might provetobefaultyincaseswheresegmentationwouldbe tough. An output layer known as Connectionist Temporal Classification(CTC) is used to overcome this issue. This kind of layer has got the ability to directly outputtheprobabilitydistributionoverlabelsequences. We normalize the output activation functions in such a way that the result is one when they are summed up. This is then treated as probability vector of the characters present at that position. The output layer contains one node for each class label plus a special node, to indicate "no character",i.e.: no decision about a character can be made at that position. Thus, there are K +1 nodes in the output layer, where K is the number of classes. The CTC objective function is defined as the negative log

probability of the network correctly labelling the entire training set. For a given training set (S) consisting of pairs of input and target sequences (x, z), the objective function O can be expressed as
$O = -\sum_{(x,z) \in S} \ln p(z|x)$.

One advantage of having such a discriminative objective function is that we can directly model the probability of the label sequence given the inputs. This has been proven better than an HMM based methods which are generative in nature [9]. Also an HMM based system assume that the probability of each observation depends only on the current state. On the other hand, a BLSTM system can easily model continues trajectories and the amount of contextual information available to such a network can in principle extend the entire input sequence.

III. LITERATURE OVERVIEW

Recognition of Printed Devanagari Text Using BLSTM Neural Network:-
In this paper, they proposed a BLSTM based system that performs recognition at word level. It results in more than 20% improvement in word accuracy while comparing traditional OCR system.

A Segmentation-Free Approach for Printed Devanagari Script Recognition:-
The complex nature of the Devanagari script (involving fused/conjunct characters) makes the OCR research in Devanagari a challenging task. They have introduced a new database, Deva-DB, comprising of Ground-truthed textline images from various scanned pages and synthetically generated text-lines. OCRopus line-recognizer has been adapted and trained on this database. This LSTM-based system yielded a character error rate of 1.2% when the test fonts matched that of the training data but the error rate increased (9%) when tested on scanned data (different set of fonts). The important issue that the network faced while classifying the characters was that of conjunct characters and the cases where characters are vertically stacked. The shape and position of these vertically stacked glyphs vary widely with different fonts. The top error is the deletion of 'c;'. To address these issues and as a future step, 2D-LSTM can be evaluated for this database and may improve the accuracy as these nets would scan the text-lines not only sideways, but also top-down. They believe that this would give an improved result since the pixel variation in the vertical direction would also be taken into account.

Devanagari font design for optical character recognition (IIT BOMBAY)
Recognition of conjuncts in the Devanagari were studied. This included the algorithm for recognition and algorithm for separation of the half form of the letter from the full form. A comprehensive testing of the font was also done and based on the result design of the characters were tweaked appropriately.

IV. CONCLUSION

The complex nature of the Hindi script (involving fused/conjunct characters) makes the OCR research in Hindi a challenging task. We proposed a BLSTM based system that performs recognition at word level. It results in more than 20% improvement in word accuracy while comparing traditional OCR system.

In future, we would like to evaluate different features for this network and also use a language model/dictionary based post-processor to improve the accuracy. We would also like to extend our work towards recognition of other Indian languages

V. REFERENCES

[1] D. Arya, T. Patnaik, S. Chaudhury, C. V. Jawahar, B.B.Chaudhuri, A.G.Ramakrishna, C. Bhagvati, and G. S. Lehal. Experiences of integration and performance testing of multilingual OCR for printed Indian scripts. In J-MOCR Workshop, ICDAR, 2011.
[2] B.B.ChaudhuriandU.Pal.AnOCRsystemtoreadtwo Indian language scripts: Bangla and Hindi. In ICDAR, 1997.
[3] S. Dutta, N. Sankaran, K. P. Sankar, and C. V. Jawahar. Robust recognition of degraded documents using character n-grams. In Document Analysis Systems, 2012.
[4] V. Frinken, A. Fischer, and H. Bunke. A novel word spotting algorithm using bidirectional long short-term memory neural networks. In ANNPR, 2010.
[5] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. IEEE Trans. Pattern Anal. Mach. Intell., 34(2), 2012.
[6] V. Govindaraju and S. Setlur. Guide to OCR for Indic Scripts. 2009.

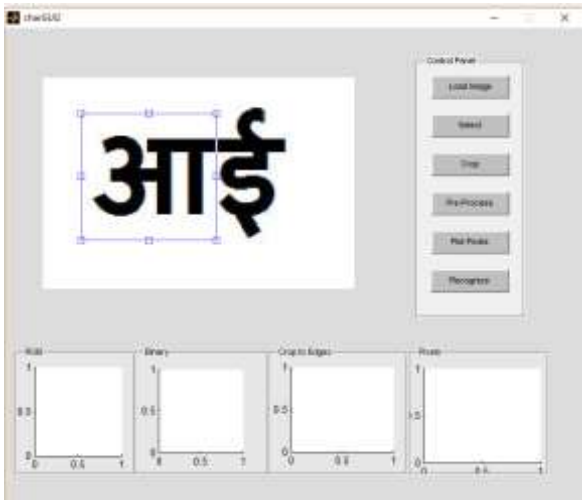
ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

[7] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell., 31(5), 2009.
[8] R. Jain, V. Frinken, C. V. Jawahar, and R. Manmatha. BLSTM neural network based word retrieval for Hindi documents. In ICDAR, 2011.
[9] P. S. Natarajan, E. MacRostie, and M. Decerbo. The bbn Byblos Hindi OCR system. In DRR, 2005.
[10] U. Pal and B. B. Chaudhuri. Indian script character recognition: a survey. Pattern Recognition, 2004

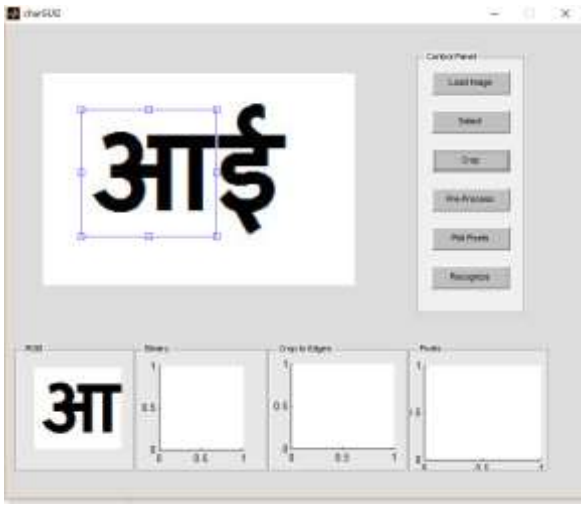
APPENDIX

After running the program, control panel appears having various functions like loading image, selecting, cropping, pre-processing, plot pixel and recognizing.

STEP 1 : Loading the image from the data set.

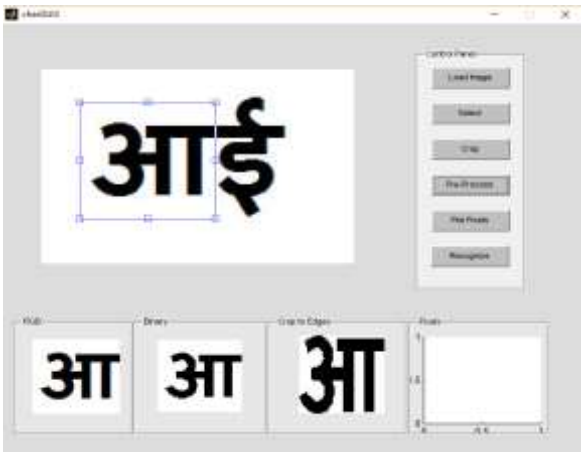


STEP 2: Selecting and cropping is done. It extracts the black part from the base through RGB.

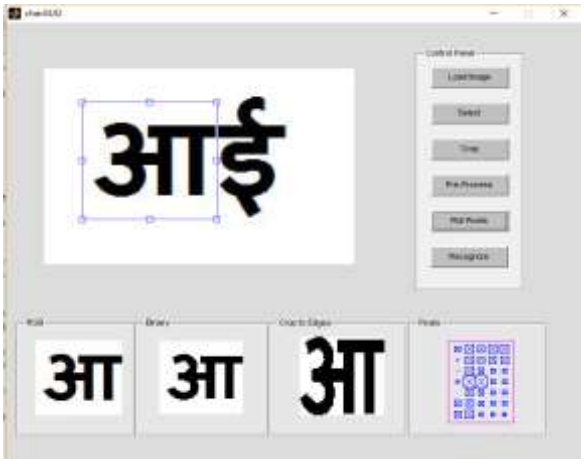


STEP 3: Pre-processing of the letter. Here binary conversion and cropping of the letter to edges is done.

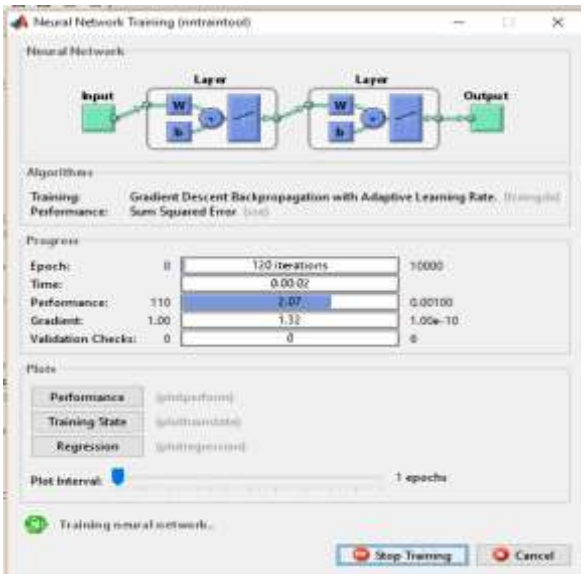
ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7



STEP 4: Here pixels of the image is calculated with Plot Pixels.



STEP 5: Recognition of the Hindi letter is done with the help of neural network algorithm.



STEP 6: After all the process is done, our image has been recognized and is displayed in the text file.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

CITE AN ARTICLE

Banger, R., Singh, M., Sharma, K., Singla, S., & Rastogi, S., Mrs. (2018). HINDI LANGUAGE RECOGNITION SYSTEM USING NEURAL NETWORKS. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, *7*(5), 98-103.