2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

A Hybrid Deep Architecture for Robust Recognition of Text Lines of Degraded Printed Documents

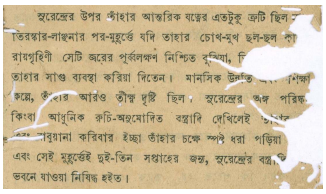
Chandan Biswas[1], Partha Sarathi Mukherjee[1], Koyel Ghosh[2], Ujjwal Bhattacharya[1], Swapan K. Parui[1]
chandanbiswas08@yahoo.com, {parthosarothimukherjee, ghosh.koyel8}@gmail.com,
{ujjwal,swapan}@isical.ac.in
[1]CVPR Unit, Indian Statistical Institute, Kolkata, India
[2]Nopany Institute of Management Studies, Kolkata, India

***Abstract*—During the last 20 years, significant research studies have been undertaken for automatic recognition of printed documents. The same is true for Bangla, a major Indian script. All these studies were mainly centered on comparatively well-behaved good quality printed documents. However, many of the large archives include significant volumes of older documents which are so degraded in their present form that they cannot be reasonably transcribed using the existing OCR (Optical Character Recognition) approaches. On the other hand, automatic recognition of printed contents of these documents has significant application potentials such as generation of descriptive metadata, full-text searching, information extraction etc. The contributions made in the present study are (i) creation of a moderately large annotated database of degraded Bangla documents towards their recognition studies, (ii) development of a Gaussian mixture model based strategy for extraction of text components from complex noisy background of such documents and (iii) development of a line level recognition scheme for degraded Bangla documents. We have studied two different CNN-BLSTM-CTC hybrid architectures for this recognition problem. The winning architecture uses the first convolution layer of the CNN in a fashion similar to the inception model of deep learning methodologies.**

I. Introduction

State-of-the-art Optical Character Recognition (OCR) methodologies are capable of providing sufficiently high accuracies on printed documents of good quality. However, their performances remain vulnerable on documents of degraded quality [1], [2]. Automatic recognition of such degraded old documents has enough application potentials towards easy management of the preservation of our cultural heritage, their content based search, indexing etc. Several factors such as worn-out condition of low quality paper, color deformation, low contrast of the text, typesetting imperfections, different shapes of alphabets due to the use of non-standard fonts etc. cause the failure of existing OCR engines on similar old documents. Thus, recognition of such documents still remains a challenging research problem in the field of document image analysis. On the other hand, certain initiatives like 'The Internet Archive', 'Digital Library of India' or the 'Million Book Project' have led to the digitization of large volumes of books many of which have undergone degradation to various extents. However, digitization alone cannot serve the intended purpose. Exploitation of these large banks of digitized knowledge to the maximum possible extent requires their automatic conversion from image into text [3].

A solution to the above problem with a limited scope can be achieved by matching keywords in degraded document images paving the way for their effective indexing towards retrieval against user-supplied query texts. In the literature, word-spotting had been studied for similar purposes [2], [4], [5]. A majority of these studies [6], [7] has dealt with old degraded manuscripts due to the extreme recognition difficulty of its handwritings of widely varying styles. Often such a study [8] is centered on some specific handwriting style. Although printed document image recognition is a much simpler problem, the existing OCR systems may perform poorly on certain printed documents [9] degraded significantly due to ageing effects or appearance of stains, non-uniform distribution of dirt caused by poor maintenance, presence of cuts or holes, non-uniform placements of characters, words or lines in old printed documents stored in archives [1]. In Fig. 1, a piece of Bangla degraded document and the output of the state-of-the art Bangla OCR engine [10], [11] on the same have been shown and the corresponding word level error is 48.65%. Earlier Bag-of-Features Hidden Markov Models were studied [12] for printed Bangla word spotting. However, to the best of our knowledge, there does not exist any recognition study of such degraded documents of Bangla.



সুরেন্দ্রর উপর,ঙ্হার আন্তরিক যত্নের এতটুক্ ত্রুটি ছিল :'
-তরন্যার০গান্থনার পর-মুহূর্ত্তে যদি তাহার চোখ-ম্মখ ছপ-ছদ ক্লা
রায়গুহিণী সেটি জ্বরের পূর্ব্বল্লাক্ষণ নিশ্চিত রুক্ষযা,।
তাহার সাগু ব্যঝন্তু করিয়া দিতেন৷ মানসিক উয়,ত ০৩৭?''ক্ষা
ঢ়ন্নে, স্তুংর আরও স্তীয় দৃষ্টি ছিল্ম, সুরেন্দ্রর অদ পরিক্ষ
' কিশুবা আধুনিক রুচি-অনুমোদিত বস্ত্রাদি দেখিশেই 'দ২
৭ 'ন০ য়াক্কয়ানা করিবার ইচ্ছা তাঁহার চক্ষে স্পণ ধরা পড়িয়া
এবং সেই মুহূর্ত্তেই দুই-তিন সপ্তাহের জন্য, সুরেত্রর বস্ত্রা:
ভবনে যাওয়া নিষিদ্ধ হইত৷

Fig. 1: State-of-the-art Bangla OCR output on a piece of degraded Bangla document.

Several recognition studies of degraded documents of Latin and a few other scripts can be found in the literature. An improved algorithm of supervised document image decoding method was studied in [13] providing high recognition accuracy even under severe image degradation. A generative probabilistic model was proposed in [14] for transcribing images of old Latin documents. A top-down segmentation based approach for recognition of both historical handwritten and printed documents was proposed in [15]. A bidirectional

978-1-5386-3787-6/18/$31.00 ©2018 IEEE

long short term memory (BLSTM) based classifier was used in [16] for recognition of degraded Devanagari words. In this method, a segmentation strategy was used for extraction of words from input document but no further segmentation of words into the constituent characters was applied and the same helped to improve the state-of-the-art accuracy of Devanagari OCR. In [17], the challenges met by open source OCR engines in recognizing documents of historical archives have been discussed while in [18], a methodology for evaluation of the performance of OCR engines on noisy historical documents had been proposed.

In the present study, we considered line level recognition of degraded printed Bangla documents to avoid possible errors during segmentation of text lines into the constituent words. We use Gaussian mixture model for extraction of texts from noisy background of similar degraded documents. A hybrid neural network architecture consisting of a convolutional neural network (CNN), two layers of BLSTM cells and a connectionist temporal classification (CTC) layer have been used for recognition of the segmented text lines. Here, we studied two different CNN-BLSTM-CTC architectures for the recognition task. In one of these two architectures, a simple CNN is used to extract the feature vector from input text line while the CNN model of the other architecture operates like Inception deep convolutional architecture which has been recently introduced as GoogLeNet in [19]. Also, we have developed a moderately large annotated image database of degraded Bangla documents for training and testing of the proposed recognition architecture. The proposed strategy improves the performance of the existing state-of-the-art Bangla OCR engine on the test set of this database.

The rest of this article is organized as follows. Some details of the document image database, called ISIDDI, have been presented in the next section.

II. Degraded document image database (ISIDDI)

The present database (ISIDDI) consists of images of 535 pages scanned from 15 old Bangla printed books. These books had been collected from three different sources, *viz.* (i) Indian Statistical Institute library, (ii) Old Book Market at College Street, Kolkata, India and (iii) the Public Library of India (https://archive.org/details/digitallibraryindia). A number of pages of the books collected from sources (i) and (ii) containing various types of degradations as discussed in [1] had been initially identified before their scanning. We scanned these degraded document pages using a flatbed scanner at 300 dpi and stored them as color images in both uncompressed TIF and JPG formats. Similarly, we have identified several pages of printed Bangla containing one or multiple types of degradations from the above archive and downloaded them. Since these books were originally written over a long period of time of the past history, both of their forms of Bangla language and font of the printed script vary widely. This dataset of printed Bangla document pages is divided into training, test and validation sets consisting of approximately 60%, 25% and 15% sample page images. There are 323, 130 and 82 sample document images in the training, test and validation sets of the ISIDDI database. Each of these samples has been ground truthed in Unicode and annotated at the line level. The numbers of word and character classes appearing in this entire database are respectively 26,663 and 320. The division of the ISIDDI into the three sets has been done randomly at the initial stage and latter some adjustments have been made such that the character lexicon size of each set becomes the same, i.e., 320. This sample database will be available (https://www.isical.ac.in/~ujjwal/download/database.html) free-of-cost for academic research purposes. A few degraded document image samples have been shown in Fig.2. A single image sample may have multiple types of degradation such as the one shown in Fig.2(h).

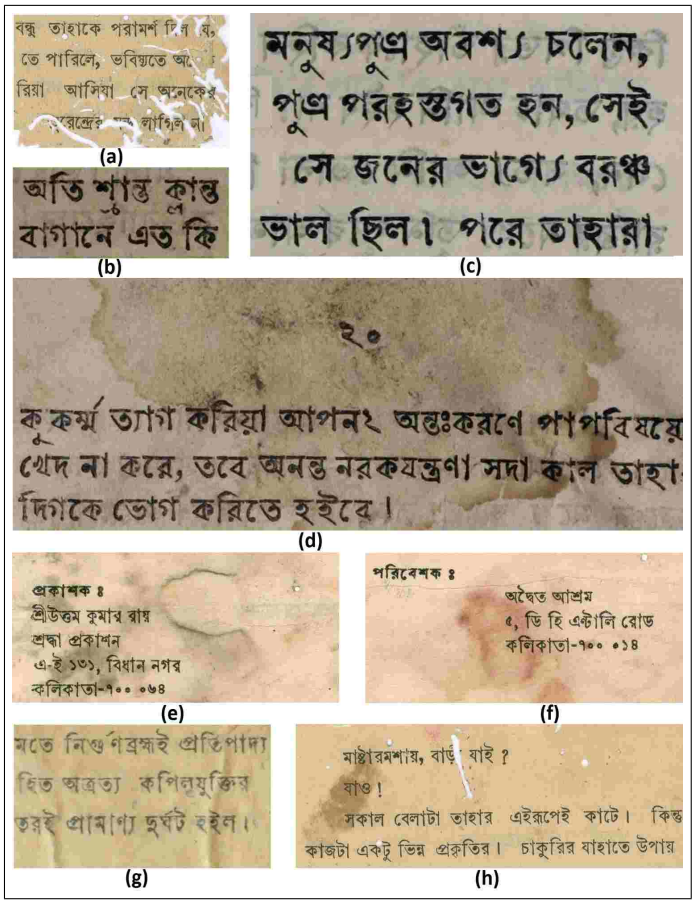
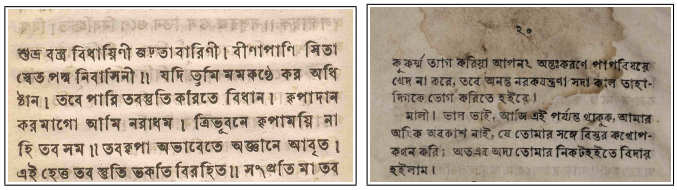


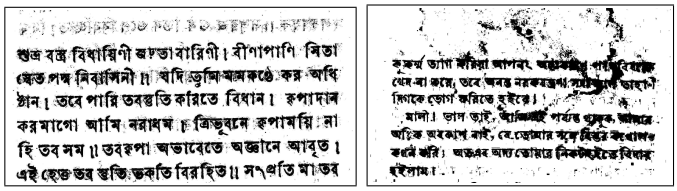
Fig. 2: A few samples of document degradation present in ISIDDI database.

III. Proposed recognition scheme

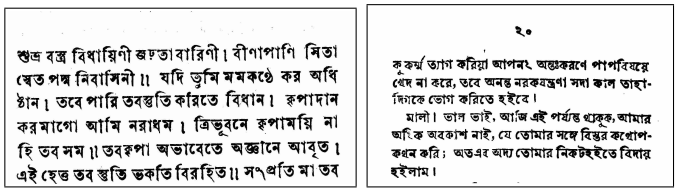
In the present study, we have developed an end to end document recognition system that accepts the image of a document page (possibly degraded quality) containing printed texts and transcribes it to the corresponding unicode strings. The work-flow of the proposed system primarily consists of two phases. The first phase consists of two parts. The sequence of preprocessing operations of the first part improves the



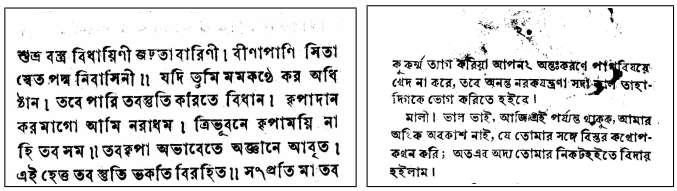
(a) Two image samples of degraded documents



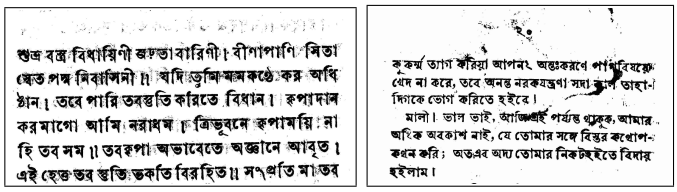
(b) Segmented by GMM with K=2



(c) Segmented by GMM with K=3 (foreground pixels of the set A_1 and pixels of the set $A_2 \cup A_3$



(d) Segmented by Otsu's method



(e) Segmented by Sauvola's method

Fig. 3: Comparative results of segmentation by different strategies for two degraded document image samples.

quality of the input image by eliminating the noise and artefact as much as possible. In the second part of the first phase, the lines in a page are segmented using an existing state-of-the-art line segmentation algorithm. These operations have been described in Section III-A.

The second phase receives a segmented line image as input from the first phase and converts it to the corresponding string of unicodes. We have developed a lexicon free recognizer and also an automatic feature extractor to achieve the desired result. For this purpose, we have constructed a hybrid network architecture consisting of CNN, RNN BLSTM and CTC layer. Details of this network architecture are provided in Sec.III-B.

A. Preprocessing stage

1) Background-Foreground segmentation: We use a Gaussian Mixture Model (GMM) for extraction of texts parts (foreground) from noisy background of a degraded document image in which each pixel has R, G, B values. We do not convert the input colour image into gray scale. The colour information is used so that the foreground can be identified in a better manner. An input image is then segmented into two and into three clusters separately in the RGB space to show that the three-clusters approach provides better results. For segmentation, a Gaussian mixture distribution is used with K Gaussian components, which is given by

$$f(\vec{x}|\lambda) = \sum_{i=1}^{K} p_i f_i(\vec{x}) \tag{1}$$

where K is either 2 or 3, $\vec{x}$ is the RGB vector of an image pixel, $f_i(\vec{x})$ i-th mixture component and p_i are the prior probabilities where $\sum_{i=1}^{K} p_i = 1$. Each f_i is a 3-dimensional Gaussian distribution as given below

$$f_i(\vec{x}) = \frac{exp\{-0.5(\vec{x}-\vec{\mu_i})^T \Sigma_i^{-1}(\vec{x}-\vec{\mu_i})\}}{\{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}\}} \tag{2}$$

where $\vec{\mu_i}$, Σ_i are the mean vector and covariance matrix of $\vec{x}$ for the i-th mixture component. Thus, the set of parameters $\lambda = (p_i, \vec{\mu_i}, \Sigma_i), i = 1, 2, ..., K$ defines the mixture of 3-dimensional Gaussian distributions given by the equation (2). Values of these parameters are estimated using the well-known Expectation Maximization (EM) algorithm [20].

After the parameters are estimated, segmentation of the input image is done as follows. A pixel is assigned to the i-th cluster (say, A_i) if $p_i f_i(\vec{x}) > p_j f_j(\vec{x})$ for $j \neq i$. Thus, we have two clusters A_1 and A_2 when $K = 2$, and three clusters A_1, A_2 and A_3 when $K = 3$. Without loss of generality, let us assume that the sum of the RGB values of the mean vector $\vec{\mu_1}$ is less than the sum of the RGB values of the mean vector $\vec{\mu_2}$. We also assume, when $K = 3$, that the sum of the RGB values of the mean vector $\vec{\mu_2}$ is less than the sum of the RGB values of the mean vector $\vec{\mu_3}$. In other words, when $K = 2$, A_1 and A_2 represent the foreground and the background respectively, and when $K = 3$, A_1 and A_3 consist of foreground and background pixels respectively while A_2 consists of pixels from both the foreground and the background and, in a sense, represents the degraded parts of the input image. For the input images shown in Fig.3(a), the segmented output images with $K = 2$ are shown in Fig.3(b). It can be observed that the noise pixels are present in both foreground and background due to the degraded nature of the input image. For the input images in Fig.3(a), the segmented output images with $K = 3$ are shown in Fig.3(c). Here the foreground is much clearer and the white part consists of pixels belonging to the set $A_2 \cup A_3$. However, since A_2 consists of both foreground and background pixels, we intend to retrieve some of these foreground pixels from A_2 to put in A_1.

Foreground-Background Separation Algorithm:

(i): Use the colour image (RGB) of the input degraded text document.

(ii): Employ EM algorithm to estimate the GMM parameters based on the RGB values of the pixels of the input image and then to generate three clusters of pixels, namely, A_1, A_2 and A_3 as defined above. A_1, A_2 and A_3 represent foreground, degraded parts and background respectively, as mentioned earlier.

(iii): If any pixel in the interior part of closed contours in the foreground part belongs to A_2, move this pixel to A_3. This is because some characters have small holes that should remain intact during Step (iv) below.

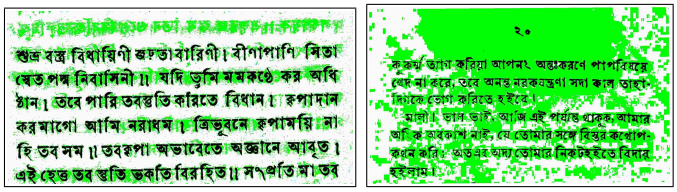
(iv): To retrieve foreground pixels from cluster A_2, for every pixel in A_1, its 8-neighbourhood is examined. If any of these 8 neighbourhood pixels belongs to A_2, include it in A_1. That is, a pixel from a degraded part is moved to the foreground. This iteration is done only once since quite often non-text pixels are regarded as text pixels from the second iteration. (See Fig.4(b)).

(v): Merge A_2 and A_3 into new A_3, i.e., the final background. A_1 already represents the final foreground. (See Fig.4(c)).

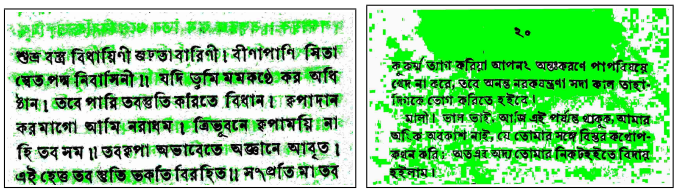
Results of foreground-background separation by the above algorithm on two document image samples of Fig. 3(a) are shown in Figs. 4(a) - 4(c) using two image samples of our ISIDDI database.

2) Skew correction of document image: Like many other Optical Characters Recognition (OCR) systems, we use a skew correction module before segmentation of lines from the document image. Often the well-known Hough transform is used for detection of lines and curves in an image which may be, in turn, used for detection of the skew present in the document image. In the present study, we have used the skew correction method of [21] based on this Hough transform to compute the skew present in the document. The foreground pixels obtained in the previous step are subjected to a rotation equal to this skew.

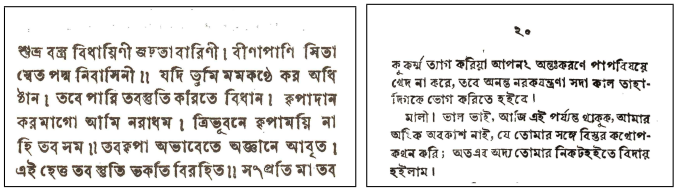
3) Line segmentation: The proposed recognition approach includes a module for segmentation of individual lines from the document image. We have used the line segmentation approach proposed in [22] to segment text lines from the skew corrected foreground-background segmented document image. This line segmentation approach is based on Hough transform and it provided 98.86% F-measure (evaluation protocol used in handwriting segmentation contest organized in conjunction with ICDAR, 2013) on the samples (training, validation and test) of ISIDDI database. The output of this line segmentation module on the input document images of Fig. 3(a) are shown in Fig. 5. In this connection, it may be noted that after obtaining the regions of individual text lines, we extract them from the foreground-background segmented image such that the segmented lines are gray level images with background pixels are completely white (gray value = 255) and text pixels



(a) Foreground (A_1), background (A_3) and degraded (A_2) pixels generated by the GMM corresponding to the image samples of Fig. 3(a) are shown in black, white and green colours respectively [Step (ii) of the above algorithm]



(b) Foreground, background and degraded pixels after executions of steps (iii) and (iv) of the above algorithm



(c) Final results of the above foreground-background separation algorithm

Fig. 4: Results of different stages of the Foreground-Background Separation Algorithm corresponding to the image samples of Fig. 3(a).

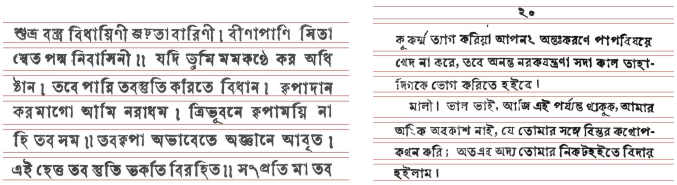


Fig. 5: Results of line segmentation on image samples of Fig.3(a)

.

can have any other gray values. These gray level line images are passed to the following module for recognition.

B. Line level recognition

The proposed recognition system is designed to accept the image of a text line and output the transcription of the same in unicode. In this context, the following issues are important.

- Transcription of the image of a text line needs either recognition of its individual words or the characters (including space). Thus, it requires a word segmentation and/or a character segmentation module. Segmentation of the words from a given line is relatively easier than that of the characters from a word. On the other hand, if

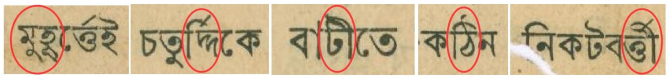


Fig. 6: A few printed Bangla words with overlapping characters (marked by red curves) posing difficulty to any character segmentation module.

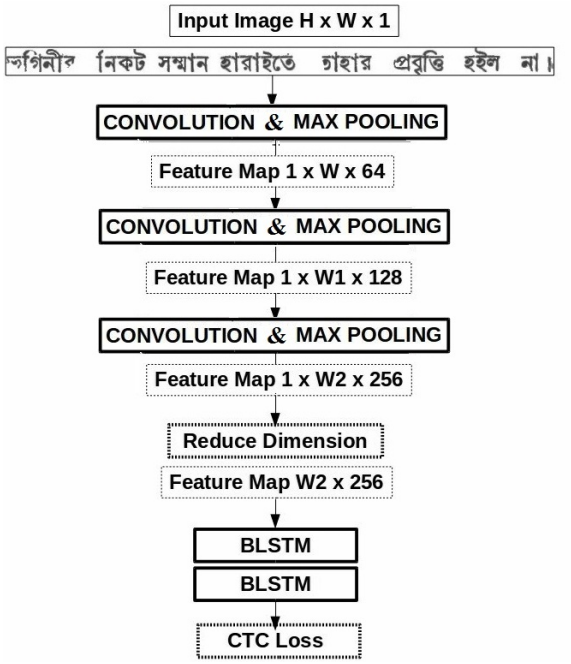


Fig. 7: CNN-BLSTM-CTC Architecture

a word classifier is employed to solve the problem, the number of classes would be very large and consequently obtaining good accuracy becomes difficult. Therefore, we need to consider a trade-off.

- Implementation of a character classifier needs to include a large number of character classes because Bangla has many modified and conjunct characters and the total number character classes in the present database is 320.
- A segmentation based recognition scheme needs prior segmentation of characters in a word (or a line). However, often two adjacent characters of a word overlap spatially (as shown in Fig. 6) and the same poses challenge to the segmentation module. The challenge is more serious for degraded documents. In the literature, various approaches [23] have been studied to address the problem. Here, we model the problem as a sequence labelling problem and used the Connectionist Temporal Classification (CTC) method [24] along with the RNN BLSTM [25], [26] classifier for the solution.
- The choice of a good *feature* set is another important issue. We use CNN for automatic feature extraction. All these together lead us to the implementation of an end-to-end segmentation free analytic recognition system for Bangla text lines similar to the approach of [27].

TABLE I: Configuration of the network architecture of Fig. 7

Layer	Size	Filters / Nodes	Stride
Convolution	223×5	64	1
Max Pooling	1×5	NA	2
Convolution	1×5	128	1
Max Pooling	1×5	NA	2
Convolution	1×5	256	1
Max Pooling	1×5	NA	2
BLSTM	NA	64	NA
BLSTM	NA	128	NA

Towards the implementation of the above, we consider two different network models. Each model is a combination of CNN and BLSTM. The first network is a simple one as shown in Fig. 7. Input to this network are segmented line images of dimensions $H \times W \times 1$ (H and W being respectively the *height* and *width* of each line image). Height of each text line is normalized to 223 pixels. Finally, length of each such text line has been adjusted to a fixed value of 2851 (the maximum length of a text line in ISIDDI database) by padding additional white (gray value = 255) pixels to its right end. The configuration of this network is presented in Table I. It should be noted that the first convolution layer has a filter having the same height as that of input images. After the convolution operation we get an array of feature maps having the shape $1 \times W \times 64$. We are considering this as images with 64 channels, width=W and height=1. So they are now as good as row images. Subsequent convolution layers map the input features to higher dimensional maps and max-pooling layers subsample the data. The output of the final max-pooling layer is fed into two consecutive BLSTM layers. This allows us to process the temporal order of the CNN output. The outcome of the BLSTM layers is processed by a CTC layer in order to find the *most probable label* against the input image.

Our second model tweaks the first CNN layer to operate like an Inception model [19]. A block diagram of this second model is shown in Fig. 8.

The motivation of simulation of this latter network model is to exploit various information extracted by using filters of multiple sizes on the input image. This is achieved by deploying multiple convolution layers in parallel. Outputs from these layers are merged along the last axis (*feature* axis). Suppose the input image has dimension $H \times W \times 1$ and there are three such convolution layers each having filters N_1, N_2 and N_3 respectively. Merging after the operation yields an array of shape $H \times W \times (N_1 + N_2 + N_3)$. The rest of the network is the same as described earlier.

Implementations of the models used for recognition of line images will be available in GitHub repository.

IV. Results And Discussions

We have simulated the proposed recognition scheme on the ISIDDI database. The first simulation is with the CNN-BLSTM-CTC architecture and the second one is with the inception style CNN-BLSTM-CTC model. In both the cases we have used RMSprop optimizer with 10^{-5} learning rate. Accuracy of the system is measured by Mean Edit Distance

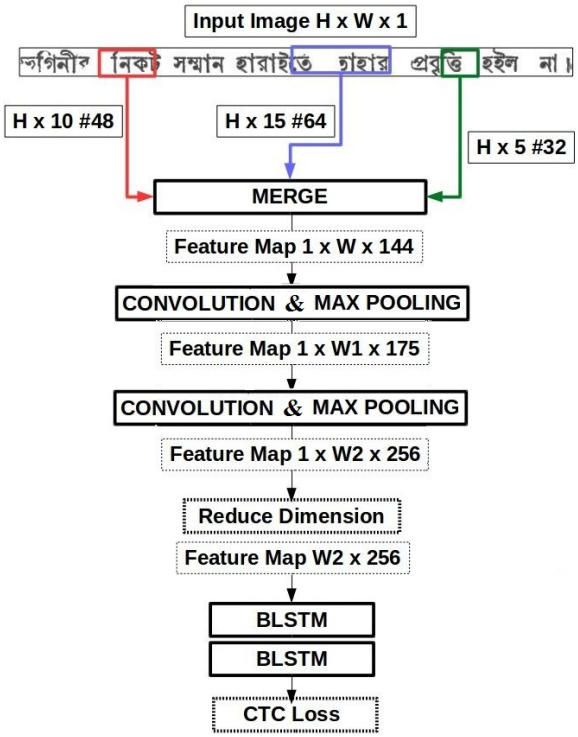


Fig. 8: Inception style CNN-BLSTM-CTC Architecture

TABLE II: Experiment Result

Model	Accuracy
CNN-BLSTM-CTC	79.38 %
Inception CNN-BLSTM-CTC	82.58 %

(Label Error Rate). In both cases network training was continued until we reach a state where 75 iterations were executed without any improvement in the validation accuracy. The results of these simulations are given in Table II. These results clearly show the superiority of the Inception style CNN-BLSTM-CTC architecture over the simple CNN-BLSTM-CTC architecture. Also, the same model provided 88.63% character level accuracy on the degraded image sample of Fig. 1 while the existing Bangla OCR system provided 81% character level accuracy on the same image sample. In future studies, we shall extend the simulation further by including a suitable language model which should increase the overall accuracy of the system.

REFERENCES

[1] A. Antonacopoulos, D. Karatzas, H. Krawczyk, and B. Wiszniewski. The lifecycle of a digital historical document: Structure and content. In *Proceedings of the 2004 ACM Symposium on Document Engineering*, DocEng '04, pages 147–154. ACM, 2004.

[2] Tomasz Adamek, Noel E. O'Connor, and Alan F. Smeaton. Word matching using single closed contours for indexing handwritten historical documents. *IJDAR*, 9(2-4):153–165, 2007.

[3] G.S. Choudhury, T. DiLauro, R. Ferguson, M. Droettboom, and I. Fujinag. Document recognition for a million books. *D-Lib Magazine*, 12(3), 2006.

[4] T. M. Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *7th International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 218–222 vol.1, 2003.

[5] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proceedings of IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages II–521–II–527, 2003.

[6] Yann Leydier, Frank Lebourgeois, and Hubert Emptoz. Text search for medieval manuscript images. *Patt. Recog.*, 40(12):3552–3567, 2007.

[7] J. A. Rodrguez-Serrano and F. Perronnin. Handwritten word image retrieval with synthesized typed queries. In *10th Int. Conf. on Document Analysis and Recognition*, pages 351–355, 2009.

[8] K. Zagoris, I. Pratikakis, and B. Gatos. Segmentation-based historical handwritten word spotting using document-specific local features. In *14th Int. Conf. on Frontiers in Handwriting Recog.*, pages 9–14, 2014.

[9] Khurram Khurshid, Claudie Faure, and Nicole Vincent. Word spotting in historical printed documents using shape and sequence comparisons. *Pattern Recognition*, 45(7):2598 – 2609, 2012.

[10] Deepak Arya, C. V. Jawahar, Chakravorty Bhagvati, Tushar Patnaik, B. B. Chaudhuri, G. S. Lehal, Santanu Chaudhury, and A. G. Ramakrishna. Experiences of integration and performance testing of multilingual OCR for printed Indian scripts. In *Proceedings of the Joint Workshop on MOCR_AND '11*, pages 9:1–9:8. ACM, 2011.

[11] B. B. Chaudhuri and U. Pal. A complete printed Bangla OCR system. *Pattern Recognition*, 31(5):531 – 549, 1998.

[12] L. Rothacker, G. A. Fink, P. Banerjee, U. Bhattacharya, and B. B. Chaudhuri. Bag-of-features hmms for segmentation-free bangla word spotting. In *Proceedings of the 4th International Workshop on Multilingual OCR*, MOCR '13, pages 5:1–5:5. ACM, 2013.

[13] Prateek Sarkar, H. S. Baird, and Xiaohu Zhang. Training on severely degraded text-line images. In *Proceedings of 7th Int. Conf. on Document Analysis and Recognition*, volume 1, pages 38–43, 2003.

[14] Taylor Berg-kirkpatrick, Greg Durrett, and Dan Klein. Unsupervised transcription of historical documents. In *Proc. of the 51st Ann. Meeting of the Asso. for Computational Linguistics*, pages 207–217, 2013.

[15] G. Vamvakas, B. Gatos, N. Stamatopoulos, and S. J. Perantonis. A complete optical character recognition methodology for historical documents. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, DAS '08, pages 525–532, 2008.

[16] N. Sankaran and C. V. Jawahar. Recognition of printed devanagari text using blstm neural network. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 322–325, 2012.

[17] Tobias Blanke, Michael Bryant, and Mark Hedges. Ocropodium: open source ocr for small-scale historical archives. *Journal of Information Science*, 38(1):76–86, 2012.

[18] Basilis Gatos Nikolaos Stamatopoulos, Georgios Louloudis. A comprehensive evaluation methodology for noisy historical document recognition techniques. In *Proceedings of The 3rd Workshop on Analytics for Noisy Unstructured Text Data*, pages 47–54. ACM, 2009.

[19] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich. Going deeper with convolutions. In *Proc. of the IEEE Conf. on Comp. Vis. and Patt. Recog.*, pages 1–9, 2015.

[20] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.

[21] Chandan Singh, Nitin Bhatia, and Amandeep Kaur. Hough transform based fast skew detection and accurate skew correction methods. *Pattern Recognition*, 41(12):3528–3546, 2008.

[22] Georgios Louloudis, Basilios Gatos, Ioannis Pratikakis, and Constantin Halatsis. Text line detection in handwritten documents. *Pattern Recognition*, 41(12):3758–3772, 2008.

[23] Yi Lu. Machine printed character segmentation; an overview. *Pattern recognition*, 28(1):67–80, 1995.

[24] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[25] S Hochreiter and J Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[26] M Schuster and K K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[27] P. S. Mukherjee, B. Chakraborty, U. Bhattacharya, and S. K. Parui. A hybrid model for end to end online handwriting recognition. In *14th International Conference on Document Analysis and Recognition (ICDAR)*, pages 658–663, 2017.