Detecting Named Entities in Unstructured Bengali Manuscript Images

Chandranath Adak*, Bidyut B. Chaudhuri†‡, Chin-Teng Lin*, Michael Blumenstein*
*Centre for AI, School of Software, FEIT, University of Technology Sydney, Australia-2007
†CVPR Unit, Indian Statistical Institute, Kolkata, India-700108
‡Techno India University, Kolkata, India-700091
Chandranath.Adak@uts.edu.au

***Abstract*—In this paper, we undertake a task to find named entities directly from unstructured handwritten document images without any intermediate text/character recognition. Here, we do not receive any assistance from natural language processing. Therefore, it becomes more challenging to detect the named entities. We work on Bengali script which brings some additional hurdles due to its own unique script characteristics. Here, we propose a new deep neural network-based architecture to extract the latent features from a text image. The embedding is then fed to a BLSTM (Bidirectional Long Short-Term Memory) layer. After that, the attention mechanism is adapted to an approach for named entity detection. We perform experimentation on two publicly-available offline handwriting repositories containing 420 Bengali handwritten pages in total. The experimental outcome of our system is quite impressive as it attains 95.43% balanced accuracy on overall named entity detection.**

***Index Terms*—Attention mechanism, Document image, Handwriting, Named Entity.**

I. Introduction

Automatic character recognition of an optically-scanned document is one of the most fascinating research areas of pattern recognition. The OCR (*Optical Character Recognition*) literature is rich, and the state-of-the-art OCR engines (e.g., ABBYY FineReader, OmniPage, PDFelement, etc.) have achieved reasonably high accuracy for printed documents. However, past OHCR (*Optical Handwritten Character Recognition*) engines [1], [2] have not fared so efficiently.

Specifically, for old and degraded documents, the existing OCR/OHCR engines may not perform well. Therefore, some alternatives are being considered because of the imminent need for document e-archiving in the digital world. In this connection, "word spotting" [3], [4] has been proposed, whereby the task is to locate the instances of a query word (keyword) residing in a document. This task is sometimes referred to as "keyword spotting" [5]. It can play a prominent role in document indexing and retrieval.

In the keyword spotting (query-by-example) task [4], the query word example is first decided by the user and then the software detects its existence in a document image. Now, in order to choose the keyword, reverse engineering may be necessary by scrutinizing the source information. Here, our research comes up with an attempt to find some keywords or important words which are not decided apriori, from a document image. Therefore, our task is clearly different from the keyword spotting.

Here, we consider the unstructured handwritten document image with an aim to find the important words, which represent date/time, place, person, organization, product, object, etc. As a matter of fact, we intend to find the entity, which represents the *name* of a living/non-living 'thing', i.e., mostly a proper noun, commonly referred to as "*named entity*".

For this named entity detection task, transcribing the textual matter with an OHCR and processing the language by an NLP (*Natural Language Processing*)-based NER (*Named Entity Recognition*) [6] model is a straight-forward approach, but it is a costly process due to the use of OHCR and NLP models. Besides, completely automated transcription of handwritten manuscripts is still risky in terms of underperformance, even when using state-of-the-art OHCR engines [1], [2]. Therefore, we strive to provide a smaller set of words that can infer the named entities, and bypass the direct use of OHCR and NLP engines. Thus, instead of machine-reading of the entire document, we reduce the search space and focus on the limited number of named entity salient words. It is also observed that the named entity form the main semantic content of a document [7]. Therefore, in this paper, we endeavor to identify handwritten words, which represent named entities and may be used to infer the content of the writing. However, the task of named entity detection from an unstructured handwritten document image without any intermediate transcript generation is quite a challenging problem. In Fig. 1, we present an example of an unstructured English handwritten sample with the named entities highlighted in bounding boxes.

In most of the Latin scripts, a named entity usually starts with an "Uppercase" character. For example, an English sentence, i.e., "*I am **F**rank*" represents *my name is Frank*. However, another sentence "*I am **f**rank*" signifies that *I am an open-minded person*. In the first sentence, "***F**rank*" (*proper noun*) denotes a person's name; however, in the second sentence, "***f**rank*" (*adjective*) denotes the characteristic of a person. Therefore, an uppercase character at the start of a word plays a crucial role in understanding named entities in Latin scripts. However, a sentence regularly starts with an uppercase character, where the corresponding word may not always be a named entity. Special care is needed to tackle this issue. In this paper, we undertake the task on a complex Indic script, *Bengali* [8], where orthographically no case (upper/lower)-dependent character shape exists. Hence,

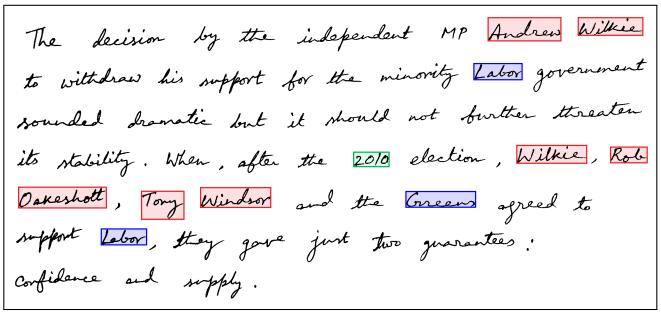


Fig. 1. Examples of named entities (person + organization + time), enclosed by boxes in an English handwritten document.

the task of named entity detection is even more challenging for Bengali script. Although our work here is on Bengali handwriting, we provide an example using the English script in Fig. 1 for easy comprehension by a mass readership. Nevertheless, some examples on Bengali script are presented in Fig. 3.

In the NLP domain, named entity identification/recognition [6] is an established problem. However, named entity detection in the field of *Document Image Processing* [9] is very new, and only a limited number of papers are presented in the literature.

The work of Zhu et al. [10] is most likely the earliest attempt to extract named entities from an *English* document image. However, the authors used a *semi-structured printed* document in the application area of "automated expense reimbursement". They combined the page layout features of the document image and the linguistic knowledge of the OCR output by a discriminative CRF (*Conditional Random Field*). On the other hand, our work emphasizes on *unstructured Bengali handwritten* manuscripts with an intention of using lexicon-free architectures.

Adak et al. [11] worked on named entity detection from unstructured *English* handwritten document images. The authors analyzed structural and positional characteristics of a word image representing a named entity. They utilized the property of English uppercase characters in the starting position of a named entity. They used object pixel distributions and projection profile-based features with a recurrent neural network to identify the named entities. However, our current work deals with the *Bengali* script where the upper/lower-case character concept does not exist. Therefore, we plan to extract auto-derived features which may reveal the latent characteristics of a named entity.

Toledo et al. [12] extracted named entities from a *Spanish* handwritten marriage record book [13]. This marriage record contained mostly the names of husband, wife, husband's father/mother, wife's father/mother with their occupations and location names. These names followed a certain arrangement which enabled the document *structured* [13]. This structure helped the authors to find semantic relationships among names. They proposed two models, (a) bigram-inspired convolutional neural network, and (b) a BLSTM (*Bidirectional Long Short-Term Memory*)-based architecture. The BLSTM-based model performed better than the other. At any rate, our work accomplishes the study on *unstructured Bengali* handwritten documents.

In essence, the objectives of our work are as follows.

(i) detecting named entities directly from unstructured Bengali handwritten document images;

(ii) bypassing the cost of the intermediate transcript generation process, as well as the cost of employing OCR/OHCR and NLP engines;

(iii) identifying the keywords/salient words/named entities that may be used to infer the content of a handwritten document.

In this paper, we propose a new convolutional architecture to extract the useful latent features for identifying named entities, which is our major contribution to this paper. Additionally, we use a BLSTM architecture followed by an attention mechanism [14] to assign weights to the extracted features. Moreover, from the application perspective, our paper provides a solution to a fairly challenging problem, i.e., named entity identification directly from unstructured Bengali manuscript images. Our work is also the earliest attempt of its kind on a non-Latin script, to the best of our knowledge.

The rest of the paper is structured as follows. The proposed method is described in *Section* II. Then, *Section* III presents the experimental results with discussions. Finally, the conclusion is drawn in *Section* IV.

II. Proposed Method

This research emphasizes the detection of words, which represent named entities, as stated earlier in *Section* I. Therefore, at first, we perform word segmentation on a given handwritten document image/page. For this purpose, we use the "*ICDAR-2013 handwriting segmentation contest*"-winning method, called "*GOLESTAN-a*" [15], which is based on a 2D Gaussian filter. Then we process further every word of a page individually to ascertain whether it is a named entity.

From a word (W), the patches ($w_{i:\ i=1\ to\ n}$) are extracted using a sliding window protocol with horizontal sliding by stride = 16. The patch size is fixed as $h \times 32$, where the patch height h is data-driven and calculated as $h = \mu_h + \alpha_h \sigma_h$; where μ_h and σ_h are the arithmetic mean and standard deviation of heights of all words in a page. Here, $\alpha_h \in \mathbb{R}$ is a tunable parameter, which works well for our task when $\alpha_h = 2$.

All the extracted patches $w_{i:\ i=1\ to\ n}$ are fed at the same time to a convolutional architecture ($ForkLinkConv$). This architecture comprises two types of modules, i.e., CCP and CC, as shown in Fig. 2. The CCP module contains two convolutional layers (conv) and one max-pool layer (pool), whereas the CC module consists of two convolutional layers (conv) only.

The output of a $\text{CCP}_{i,j}$ module *forks*, and is input to the $\text{CCP}_{i,j+1}$ and $\text{CC}_{i,j}$ module; the input of a $\text{CCP}_{i+1,j+1}$ module is the *link* (addition) of the output of $\text{CCP}_{i+1,j}$ and the output of $\text{CC}_{i,j}$ module; $\forall i = 1\ to\ n-1, \forall j = 1\ to\ 3$. The input of the $\text{CCP}_{i,1}$ is w_i; $\forall i = 1\ to\ n$. The input of $\text{CCP}_{1,j+1}$

is the output of $\text{CCP}_{1,j}$; $\forall j = 1\ to\ 3$. Here, we coin the term "ForkLinkConv" to call our convolutional architecture, and present it in Fig. 2. The i and j indices can be tallied from the horizontal and vertical sequences of CCP and CC modules of Fig. 2.

In ForkLinkConv, a CC module typically passes the residual feature learned from a CCP module of the previous patch. This technique can be beneficial for our task due to its adaptive learning and information sharing characteristics [16]. In this architecture, we use kernels of the same size for all convolutions, i.e., 3×3; and it is 2×2 for all max-pooling operations. The employed convolutional stride is equal to 1 and the pooling stride is 2. The size of zero-padding is set to 1 for both convolution and pooling operations. In Fig. 2, we mention the number of kernels used for the convolution with '$@k$', above each of the CCP and CC modules. Here, the number of kernels in $\text{CCP}_{i,j}$, $\text{CCP}_{i+1,j}$ and $\text{CC}_{i,j}$ are the same; $\forall i = 1\ to\ n$, $\forall j = 1\ to\ 4$. The output volume of a layer depends on the size of zero-padding, kernel size, stride value and the number of kernels. After each convolutional layer, we use the *leaky ReLU* (*Rectified Linear Unit*) [17] to prevent neurons from "dying". A *dropout* [18] of 50% is applied at the end of the ForkLinkConv after the full connection to alleviate the overfitting problem. From this ForkLinkConv, each of the w_i patches produces a 1024-dimensional feature vector x_i.

$$x_i = ForkLinkConv(w_i); \quad \forall i = 1\ to\ n \tag{1}$$

Now the embedding is fed to a BLSTM (*Bidirectional Long Short-Term Memory*)-based encoder for further conversion of the feature vector sequence in order to capture the contextual information. For the BLSTM encoder, the activation vectors are obtained as $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$, where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are forward and backward hidden activation vectors, respectively, i.e.,

$$\overrightarrow{h_t} = LSTM(\overrightarrow{h_{t-1}}, x_t; \lambda_E^F); \quad \overleftarrow{h_t} = LSTM(\overleftarrow{h_{t+1}}, x_t; \lambda_E^B); \tag{2}$$

where, x_t is a ForkLinkConv feature; λ_E^F and λ_E^B are tunable encoder parameters. In our task, the BLSTM layers contain 512 memory cells.

Now, we ascertain the *attention* mechanism [14] by employing attention weights in order to emphasize relevant patch information. In this task, we use *soft attention* [19] to train with gradient descent. The attention weight value a_i is calculated as follows.

$$a_i = softmax(q_i); \quad \forall i = 1\ to\ n \tag{3}$$

where, $q_i = v_A^\intercal tanh(\omega_A h_t^{(i)} + b_A)$; $a_i \in \mathbb{R}^n$; $\sum_{i=1}^{n} a_i = 1$; $h_t^{(i)} \in \mathbb{R}^{1024}$; and v_A, b_A are vectors, and ω_A is a weight matrix, which can be tuned.

The patch-wise multiplication of the attention weight a_i and ForkLinkConv-extracted features x_i produces the final feature representation l_i.

$$l_i = a_i.x_i; \quad \forall i = 1\ to\ n \tag{4}$$

At this point, the l_i feature of a patch w_i is fed to a *Fully Connected* layer (FC) that has d number of neurons, where d

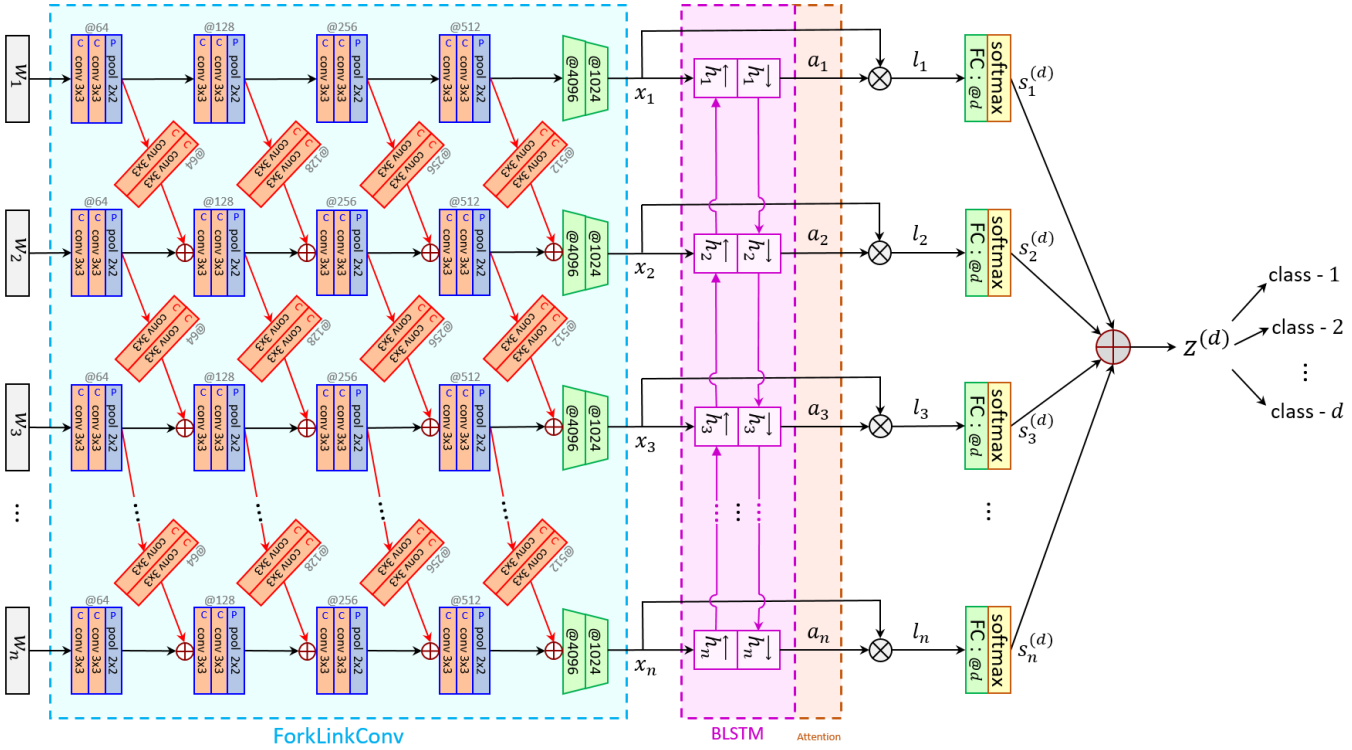


Fig. 2. Our proposed architecture.

is equal to the number of output classes. Then for each patch, a softmax layer is added to produce the probability distribution $s_i^{(d)}$ over class labels, where $\sum_d s_i^{(d)} = 1$.

Finally, the classification decision is taken from the decision rule $z^{(d)}$, which is the weighted sum of $s_i^{(d)}$ over all the patches. Here, we use an equal weight $\frac{1}{n}$, therefore it turns into the average, i.e.,

$$z^{(d)} = \frac{1}{n}\sum_{i=1}^{n} s_i^{(d)}; \tag{5}$$

where, $\sum_d z^{(d)} = 1$.

In this task, we adopt an improved version of cross entropy loss, i.e., *focal loss* ($\mathcal{L}$) [20] due to its boosting performance for imbalanced data [21]. The $\mathcal{L}$ is computed as follows.

$$\mathcal{L} = -(1 - z^{(d)})^{\gamma} \log(z^{(d)}); \tag{6}$$

where, γ is a tunable parameter which works best for our task when it equals to 2.

It can be noted that usually, the presence of named entities is quite low in regular text. Therefore, the database may be imbalanced for our task. Here, we define a term, called *named entity* (*NE*) *rate* which is the proportion of NE count and the total number of words in a database.

$$NE\ rate = \frac{NE\ count}{total\ number\ of\ words} \tag{7}$$

As a matter of fact, the *NE rate* is quite low for an imbalanced database.

III. Experiments and Discussion

In this section, at first, we discuss the database we employed and then evaluate the performance of our system through experiments.

A. Database Employed

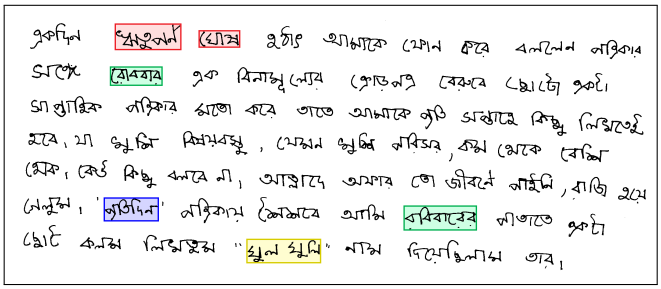
For experimentation, we used two databases containing Bengali unstructured handwritten samples. From these handwritten samples, we generated the ground-truth by labeling the named entities. Here, the named entities were basically the proper nouns, according to the Bengali grammar. In Fig. 3, we present some samples used for the experiments.

The brief details of the databases employed are as follows.

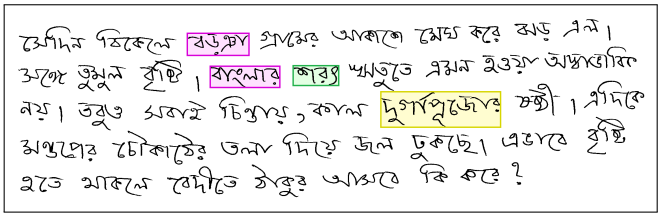
(i) DB_{PBOK} : We used 120 Bengali handwritten pages from the *PBOK-Bangla* database [22] for our experiments. Here, the total number of words was 9311 and the total count of named entities (NE) was 613. Therefore, the *NE rate* became $6.58\%(= \frac{613}{9311} \times 100\%)$.

(ii) DB_{ISI} : The *NewISIdb:HwP_Basic* database [23] contained 300 Bengali handwritten pages, which we used for the experimental analysis. A total of 30600 words was present in this database, and the total number of named entities was 2400. Here, the *NE rate* was 7.84%.

The NE rates of DB_{PBOK} and DB_{ISI} are very low due to the imbalanced nature [21] of the databases, where the NE count is substantially less than the non-NE count. The "non-NE" refers to a word that is not a named entity. The non-NE count is simply the subtraction of NE count from the total word count in a database.



(a) Sample from DB_{PBOK} [22].



(b) Sample from DB_{ISI} [23].

Fig. 3. Bengali handwritten samples with highlighted named entities, referring to person (P), organization (G), location (L), time/day/season (T), object/other non-living things (O).

TABLE I
Employed Database Details: Named Entity Count

Category	DB_{PBOK}	DB_{ISI}	DB_{PBOK} + DB_{ISI}
P: Person	459	600	1059
G: Organization	55	0	55
L: Location	0	600	600
T: Time	33	300	333
O: Object	66	900	966
# total NE*	613	2400	3013
# total word	9311	30600	39911
NE rate (%)	6.58	7.84	7.55

* NE: Named Entity

In TABLE I, we present the quantitative details of the databases employed with the frequencies of named entity categories. Here, we provide the counts of the individual named entities which refer to names of a person (**P**), organization (**G**), location (**L**), time/day/month/year/season (**T**), object/other non-living things (**O**).

The databases were divided into *training*, *validation*, and *test* sets with a ratio of 3:1:2. Moreover, the training data was augmented by adding Gaussian noise and salt & pepper noise to prevent overfitting.

B. Results and Evaluation

In our architecture, we used a gradient-based optimization algorithm, i.e., *RMSprop* [24], since it outperformed some other optimization methods such as *ADADELTA* [25], *Adam* [26], etc., when compared with respect to our task. Here, we used *momentum* = 0.9 with *learning rate* = 0.001 and *learning rate decay* = 10^{-6}. To accelerate the training speed efficiency, we employed *batch normalization* [27]. The training

was performed with up to 20K *iterations* while the *mini batch size* was set to 32. The parameters of our architecture were tuned on the *validation* set. We present the results here by testing on the *test* set.

We performed our experiments in two separate setups, as described below.

***(i) Setup-1* :** Here, we categorized the named entities in five different groups, i.e., person (**P**), organization (**G**), location (**L**), time (**T**), and object (**O**). Besides, the non-NE words created an additional group. Therefore, in this setup, the task became a *multiclass* classification, i.e., classifying into six classes.

***(ii) Setup-2* :** In this setup, all the named entity categories were united together to represent a single group. Here, we basically separated the named entities and non-NE words. Therefore, at this point, the task was perceived as a *binary* classification to classify two groups, i.e., NE versus non-NE. In fact, this setup executed the overall named entity detection.

In Fig. 4, we present three confusion matrices generated from the experimental *Setup-1*, when the system is executed on the DB_{PBOK}, DB_{ISI}, and DB_{PBOK}+DB_{ISI} databases. From these confusion matrices, we evaluate the whole system performance in terms of *accuracy* (%) and it is illustrated in TABLE II. The system produces a better result for database DB_{ISI} than DB_{PBOK}. Combining these two databases (DB_{PBOK}+DB_{ISI}), the system yields a 96.06% accuracy.

In TABLE III, we exhibit the results of experimental *Setup-2*. Here, we show the overall performance of named entity detection. For this performance analysis, in addition to the general *accuracy* measure, we calculate the *balanced accuracy* [28]. For our imbalanced data, the balanced accuracy evaluates

(a)

DB_{PBOK}	Actual: P	G	L	T	O	non-NE
P	134	3	0	0	3	16
G	8	13	0	0	2	38
L	0	0	0	0	0	0
T	1	0	0	8	0	22
O	4	0	0	1	14	37
non-NE	6	2	0	2	3	2786

(Predicted category)

(b)

DB_{ISI}	Actual: P	G	L	T	O	non-NE
P	165	0	6	2	12	59
G	0	0	0	0	0	0
L	9	0	174	5	15	43
T	2	0	3	82	4	18
O	11	0	8	5	251	102
non-NE	13	0	9	6	18	9178

(Predicted category)

(c)

DB_{PBOK} + DB_{ISI}	Actual: P	G	L	T	O	non-NE
P	304	2	8	5	15	77
G	0	13	2	0	6	43
L	13	1	169	6	13	58
T	2	0	3	84	5	25
O	13	0	7	7	263	149
non-NE	21	2	11	9	20	11947

(Predicted category)

Fig. 4. Confusion matrices of the category-labeled named entities for experimenting on (a) DB_{PBOK}, (b) DB_{ISI} and (c) DB_{PBOK} + DB_{ISI} databases.

TABLE II
NAMED ENTITY CATEGORIZATION

Database	*Accuracy* (%)
DB_{PBOK}	95.23
DB_{ISI}	96.56
DB_{PBOK} + DB_{ISI}	96.06

TABLE III
OVERALL NAMED ENTITY DETECTION

Database	*Accuracy* (%)	*Balanced Accuracy* (%)
DB_{PBOK}	95.93	94.86
DB_{ISI}	97.37	95.94
DB_{PBOK} + DB_{ISI}	96.88	**95.43**

the system efficiently. The balanced accuracy is actually the arithmetic mean of *sensitivity* (i.e., true positive rate) and *specificity* (i.e., true negative rate) measure of the system.

$$balanced\ accuracy = \frac{sensitivity + specificity}{2} \quad (8)$$

Here also, the DB_{ISI} performs better than DB_{PBOK}. The system obtains a 95.43% balanced accuracy for overall named entity detection.

C. Limitation

We observed a drawback of our system. Sometimes, a single word is disrupted due to insufficient space being left in a text line. Therefore, sometimes, a writer breaks the word into two parts, and writes one part at the end of a text-line and another part at the beginning of the consecutive text-line. If such an interrupted word is actually a named entity, then our system is unable to detect this. In Fig. 5, we present an example of such a scenario, where our system misses to detect an interrupted named entity. However, such a scenario may be overcome by making our system independent of the word segmentation module.

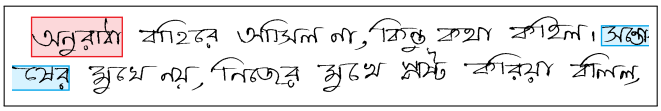


Fig. 5. An example [22] of our system limitation; *Hit*: uninterrupted NE and *Miss*: interrupted NE, highlighted in red and cyan, respectively.

D. Comparison

As discussed in *Section* I, very few studies exist in the literature related to our task. Among those, the initial work of Zhu et al. [10] was on semi-structured *printed* documents and sought assistance from an *OCR engine*, which contrasts with our objective.

Adak et al. [11] worked on *English* handwriting and relied heavily on the uppercase character at the starting position of a named entity. Since we work on *Bengali* handwriting, this previous method [11] cannot be adapted due to the absence of upper/lower-case character concept in Bengali script.

TABLE IV
COMPARISON: NAMED ENTITY DETECTION

Method	*Balanced Accuracy* (%)
Toledo et al. [12]: BLSTM	86.74
Proposed	**95.43**

Toledo et al. [12] worked on *structured Spanish* handwritten marriage records. Their best performing model is based on a BLSTM architecture and is quite generic, therefore it can be compared to our method. We adopted their BLSTM-based model and executed it on our Bengali database. For comparative analysis, we chose our experimental *Setup-2*, i.e., binary classification (NE versus non-NE). The comparison is performed on overall named entity detection, while experimenting on the total database, i.e., DB_{PBOK}+DB_{ISI}. The comparative measure in terms of balanced accuracy is shown in TABLE IV, where our system performance is superior. The best performing model of Toledo et al. [12] obtained an 86.74% balanced accuracy, while ours was 95.43%. Moreover, our method can handle *unstructured* documents.

To the best of our capacity to search the literature, we did not find any similar work on Bengali/Indic script, even on a non-Latin script, for comparison purposes.

IV. CONCLUSION

In this paper, we detect named entities directly from unstructured handwritten document images. We propose a new convolutional architecture which can extract some latent features from the text images. These features are embedded in a BLSTM network and receive weights by an attention mechanism. Our system does not require any intermediate transcription generation stage, which is the main advantage of this research. We perform our task on Bengali script, which is very complex in nature for named entity detection, due to the nonexistence of a concept similar to upper/lower-case Latin characters. We have executed our experiments on two Bengali handwriting databases and obtained an overall balanced accuracy of 95.43%. Our next endeavor will be to upgrade the system performance and to test it on some other scripts. In the future, we will also attempt to understand the semantic relationships among the extracted named entities.

ACKNOWLEDGMENT

We thank all the linguistic and handwriting experts that were consulted for their valuable input.

REFERENCES

[1] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition", IEEE Trans. on PAMI, vol. 31, no. 5, pp. 855-868, 2009.

[2] R. Plamondon, S. N. Srihari, "Online and Off-line Handwriting Recognition: A Comprehensive Survey", IEEE Trans. on PAMI, vol. 22, no. 1, pp. 63-84, 2000.

[3] R. Manmatha, C. Han, E. M. Riseman, "Word Spotting: A New Approach to Indexing Handwriting", Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 631-637, 1996.

[4] A. P. Giotis, G. Sfikas, B. Gatos, C. Nikou, "A Survey of Document Image Word Spotting Techniques", Pattern Recognition, vol. 68, pp. 310-332, 2017.

[5] D. Aldavert, M. Rusiol, R. Toledo, J. Llads, "A Study of Bag-of-Visual-Words Representations for Handwritten Keyword Spotting", Int. Journal on Document Analysis and Recognition (IJDAR), vol. 18, no. 3, pp. 223-234, 2015.

[6] D. Nadeau and S. Sekine, "A Survey of Named Entity Recognition and Classification", Journal Lingvisticae Investigationes, vol. 30, no. 1, pp. 3-26, John Benjamins Pub. Co., 2007.

[7] G. D. Zhou, J. Su, "Named Entity Recognition using an HMM-based Chunk Tagger", Proc. Annual Meeting on Association for Computational Linguistics (ACL), pp. 473-480, 2002.

[8] T. Bagchi, "Bengali Writing", Sec. 34 in *The World's Writing Systems*, P. T. Daniels, W. Bright eds., Oxford University Press, NY, 1996.

[9] D. Doermann, K. Tombre eds., "Handbook of Document Image Processing and Recognition", Springer-Verlag London, 2014.

[10] G. Zhu, T. J. Bethea, V. Krishna, "Extracting Relevant Named Entities for Automated Expense Reimbursement", Proc. ACM Conf. on Knowledge, Discovery and Data Mining (KDD), pp.1004-1012, 2007.

[11] C. Adak, B. B. Chaudhuri, M. Blumenstein, "Named Entity Recognition from Unstructured Handwritten Document Images", Proc. Int. Workshop on Document Analysis Systems (DAS), pp. 375-380, 2016.

[12] J. I. Toledo, M. Carbonell, A. Forns, J. Llads, "Information Extraction from Historical Handwritten Document Images with a Context-aware Neural Model", Pattern Recognition, vol. 86, pp. 27-36, 2019.

[13] V. Romero, A. Forns, N. Serrano, J. A. Snchez, A. H. Toselli, V. Frinken, E. Vidal, J. Llads, "The ESPOSALLES Database: An Ancient Marriage License Corpus for Off-line Handwriting Recognition", Pattern Recognition, vol. 46, no. 6, pp. 1658-1669, 2013.

[14] A. Vaswani et al., "Attention Is All You Need", Proc. Neural Information Processing Systems (NIPS), pp. 6000-6010, 2017.

[15] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, A. Alaei, "ICDAR 2013 Handwriting Segmentation Contest", Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), pp. 1402-1406, 2013.

[16] S. He, L. Schomaker, "Deep Adaptive Learning for Writer Identification Based on Single Handwritten Word Images", Pattern Recognition, vol. 88, pp. 64-74, 2019.

[17] A. L. Maas, A. Y. Hannun, A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models", ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

[18] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", Proc. Int. Conf. on Machine Learning (ICML), pp. 2048-2057, 2015.

[20] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollr, "Focal Loss for Dense Object Detection", Proc. Int. Conf. on Computer Vision (ICCV), pp. 2999-3007, 2017.

[21] H. He, E. A. Garcia, "Learning from Imbalanced Data", IEEE Trans. on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, 2009.

[22] A. Alaei, U. Pal, P. Nagabhushan, "Dataset and Ground Truth for Handwritten Text in Four Different Scripts", Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI), vol. 26, no. 04, #1253001, 2012.

[23] C. Adak, B. B. Chaudhuri, M. Blumenstein, "Offline Cursive Bengali Word Recognition Using CNNs with a Recurrent Model", Proc. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp. 429-434, 2016.

[24] T. Tieleman, G. Hinton, "Lecture 6e - RMSprop: Divide the Gradient by a Running Average of Its Recent Magnitude", Coursera: Neural Networks for Machine Learning, 2012.

[25] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method", arXiv:1212.5701, 2012.

[26] D. P. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization", arXiv:1412.6980, 2014.

[27] S. Ioffe, C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv:1502.03167, 2015.

[28] K. H. Brodersen, C. S. Ong, K. E. Stephan, J. M. Buhmann, "The Balanced Accuracy and Its Posterior Distribution", Proc. Int. Conf. on Pattern Recognition (ICPR), pp. 3121-3124, 2010.